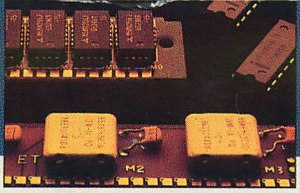
MEMORY ARCHITECTURES▶



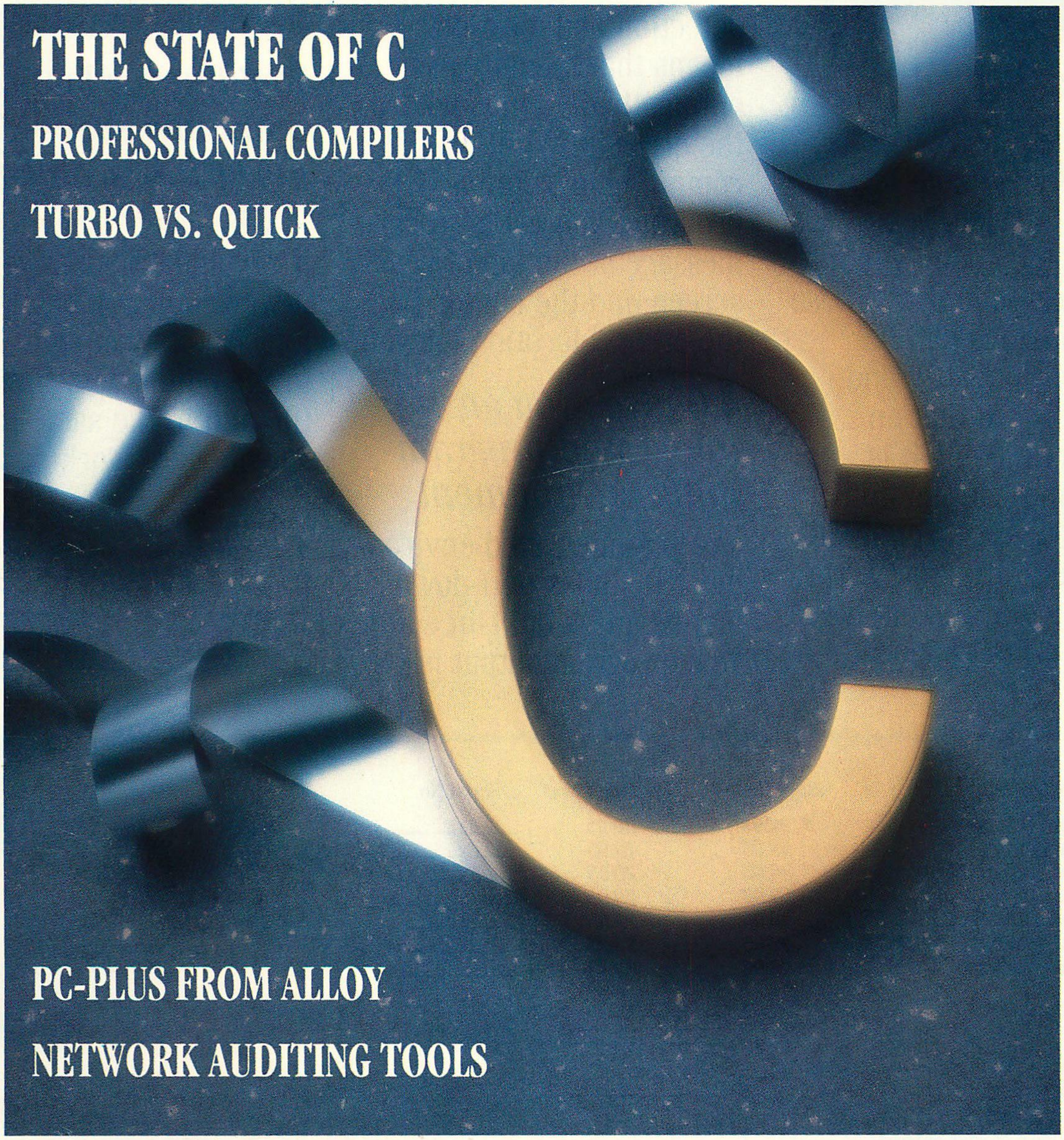
MULTIUSER DATA MANAGERS

FEBRUARY 1988

VOL. 6 NO. 2 $3.95

PC TECH JOURNAL®

FOR SYSTEMS DEVELOPERS AND INTEGRATORS



02

0 71486 02166 7

The fast lane is *fas*

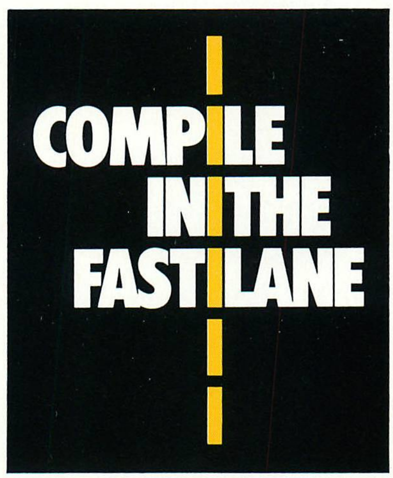
Our new Turbo Pascal® 4.0 is so fast, it's almost reckless. How fast? Better than 27,000 lines of code per minute. That's much faster than 3.0 or any other Pascal compiler and the reason why you need 4.0 today.

Pascal. The fastest and the best.

If you're just now learning a computer language, learn Pascal. If you're already programming in Pascal, you're programming with a winner because Pascal is the worldwide language of choice. Pascal is the most popular language in university computer science classes and with computer enthusiasts who appreciate Pascal's modern programming structure. It's powerful, coherent, easy to learn and use—and with Turbo Pascal 4.0—faster than ever before.

Turbo Pascal: Technical excellence

Commitment to technical excellence and



superiority also means commitment to detail, however painstaking, and that takes time. 4.0's predecessor, Turbo Pascal 3.0 is the worldwide standard, and with Turbo Pascal 4.0, we've bettered that standard. 4.0 is clearly the world's fastest development tool for the IBM® PS/2 series, PC's and compatibles—and the world's favorite Pascal compiler.

4.0 breaks the code barrier

No more swapping code in and out to beat the 64K code barrier. Designed for large programs, Turbo Pascal 4.0 lets you use all 640K memory in your computer. You paid for all that memory, now you can use it freely.

For the IBM PS/2 and the IBM and Compaq families of personal computers and all 100% compatibles.

YES! **I want to upgrade to Turbo Pascal 4.0 and the 4.0 Toolboxes**

Registered owners have been notified by mail. If you are a registered Turbo Pascal user and have not been notified of Version 4.0 by mail, please call us at *(800) 543-7543*. To upgrade if you have not registered your product, just send the original registration form from your manual and payment with this completed coupon to:

Pascal 4.0 Upgrade Dept.
Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066

Name ____________________________

Ship Address ________________________

City __________________ State ______

Zip __________ Telephone (   ) __________

PTJ 2/88

This offer is limited to one upgrade per valid registered product. It is good until June 30, 1988. Not good with any other offer from Borland.
Outside U.S. make payments by bank draft payable in U.S. dollars drawn on a U.S. bank. CODs and purchase orders will not be accepted by Borland.

ter than ever before!

4.0 uses logical units for separate compilation

Pascal 4.0 lets you break up the code gang into "units," or "chunks." These logical modules can be worked with swiftly and separately—so that an error in one module is seeable and fixable, and you're not sent through all your code to find one error. Compiling and linking these separate units happens in a flash because your compiling horsepower is better than 27,000 lines a minute.* And 4.0 also includes an automatic project Make.

4.0's cursor automatically lands on any trouble spot

4.0's interactive error detection and location means that the cursor automatically lands where the error is. While you're compiling or running a program, you get an error message at the top of your screen *and* the cursor flags the error's location for you.

4.0 gives you an integrated programming environment

4.0's integrated environment includes pull-down menus and a built-in editor. Your program output is automatically saved and shown in the output window. You can Scroll, Pan, or Page through all your output and know where everything is all the time. Given 4.0's integration, you can edit, compile, find and correct errors—all from inside the integrated development environment.

You'll never lose your mind, because 4.0 never loses your place

Whenever you re-load 4.0, it remembers what you and it were doing before you left. It puts you right back in the editor with the same file and in the same place as you were working last.

*Run on an 8 MHz IBM AT.

**If within 60 days of purchase this product does not perform in accordance with our claims, call our customer service department, and we will arrange a refund.

Please check box(es)	*Sugg. Retail*	*Upgrade Price†*	Serial No.
☐ Turbo Pascal 4.0 Compiler	*$ 99.95*	*$ 39.95*	________
☐ Turbo Pascal Tutor	*69.95*	*19.95*	________
☐ Turbo Pascal Database Toolbox	*99.95*	*29.95*	________
☐ Turbo Pascal Graphix Toolbox	*99.95*	*29.95*	________
☐ Turbo Pascal Editor Toolbox	*99.95*	*29.95*	________
☐ Turbo Pascal Numerical Methods Toolbox	*99.95*	*29.95*	________
☐ Turbo Pascal Gameworks	*99.95*	*29.95*	________

Total product amount *$* ________

CA and MA residents add sales tax *$* ________

In US please add $5 shipping and handling for each product ordered *$* ________

Outside US please add $10 shipping and handling for each product ordered *$* ________

Total amount enclosed *$* ________

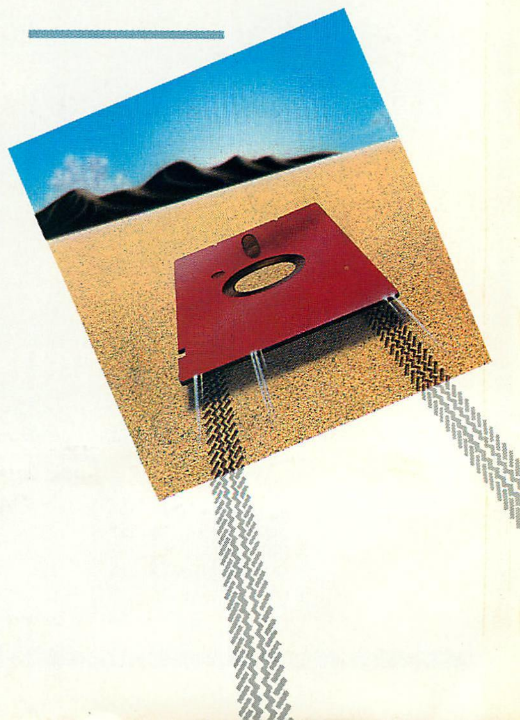
Please specify diskette size: ☐ 5¼″ ☐ 3½″

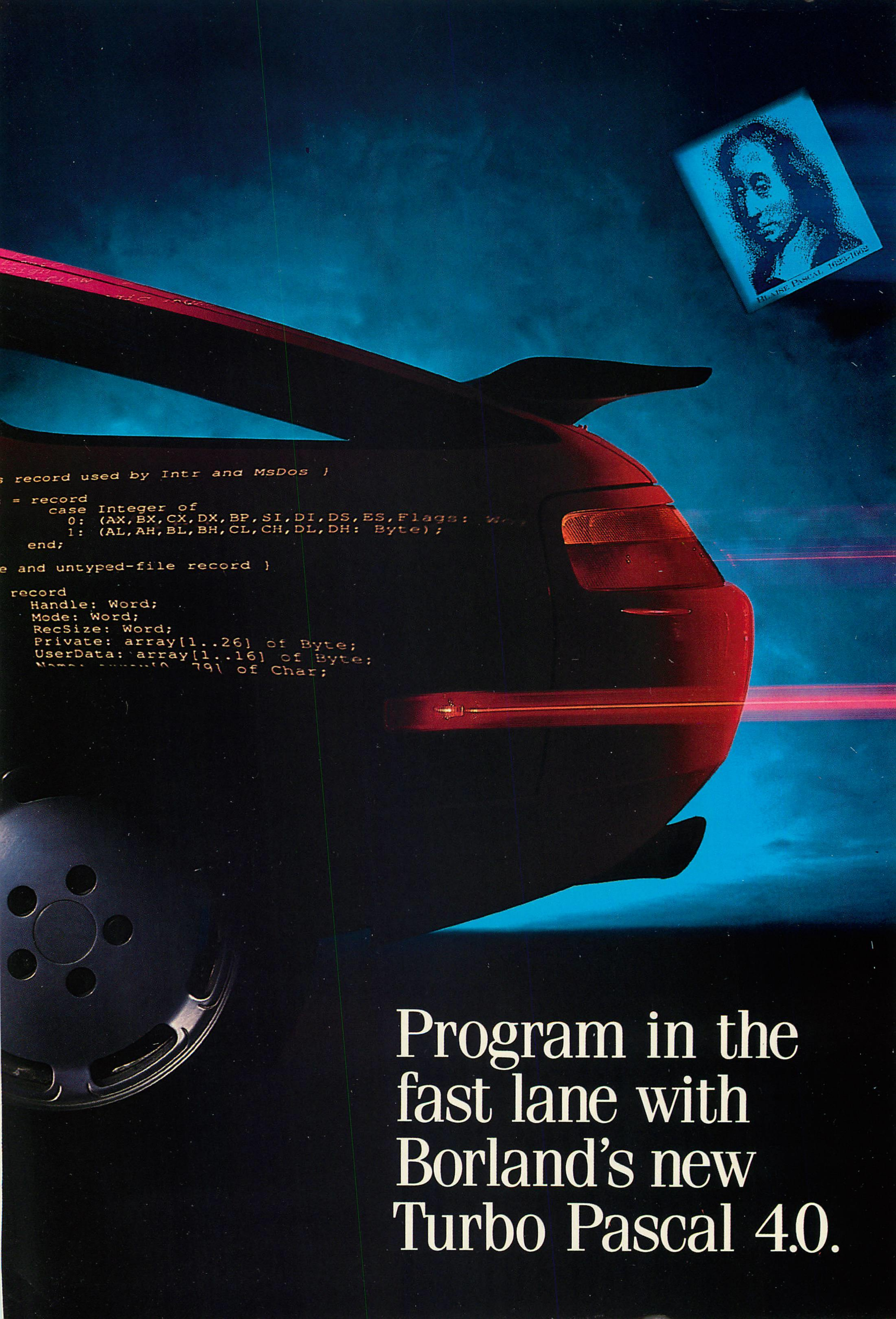
Payment: ☐ VISA ☐ MC ☐ Check ☐ Bank Draft

Credit card expiration date: ________/________

Card # |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

†To qualify for the upgrade price you must give the serial number of the equivalent product you are upgrading. PTJ 2/88




Blaise Pascal 1623-1662
record used by Intr and MsDos }
= record
case Integer of
0: (AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags:
1: (AL,AH,BL,BH,CL,CH,DL,DH: Byte);
end;
and untyped-file record }
record
Handle: Word;
Mode: Word;
RecSize: Word;
Private: array[1..26] of Byte;
UserData: array[1..16] of Byte;
of Char;
Program in the fast lane with Borland's new Turbo Pascal 4.0.

Now's the time for a *fast* decision: Upgrade now to 4.0!

Compatibility with Turbo Pascal 3.0

We've created 4.0 to be highly compatible with version 3.0 and included a conversion program and compatibility units to help you convert all your 3.0 programs to 4.0.

Highlights of Borland's new Turbo Pascal 4.0

- Compiles 27,000 lines per minute
- Supports >64K programs
- Uses units for separate compilation
- Integrated development environment
- Interactive error detection/location
- Includes a command line version of the compiler

4.0 also

- Saves output screen in a window
- Supports 25, 43 and 50 lines per screen
- Generates MAP files for debugging
- Has graph units including CGA, EGA, VGA, MCGA, 3270 PC, AT & T 6300 & Hercules support
- Supports extended data types (including word, long integers)
- Does smart linking
- Comes with a free revised MicroCalc spreadsheet source code

4.0 is all yours for only $99.95

Sieve (25 iterations)

	Turbo Pascal 4.0	**Turbo Pascal 3.0**
Size of Executable File	2224 bytes	11682 bytes
Execution speed	9.3 seconds	9.7 seconds

Sieve of Eratosthenes, run on an 8MHz IBM AT

Since the source file above is too small to indicate a difference in compilation speed we compiled our GOMOKU program from Turbo Gameworks to give you a true sense of how much faster 4.0 really is!

Compilation of GO.PAS (1006 lines)

	Turbo Pascal 4.0	**Turbo Pascal 3.0**
Compilation speed	2.2 seconds	3.6 seconds
Lines per minute	27,436	16,750

GO.PAS compiled on an 8 MHz IBM AT

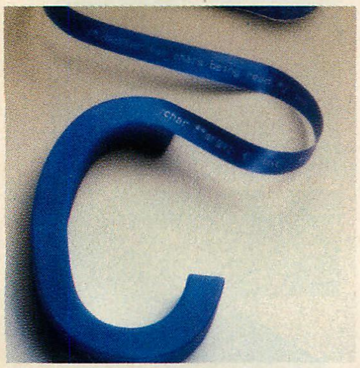
60-Day Money-Back Guarantee**

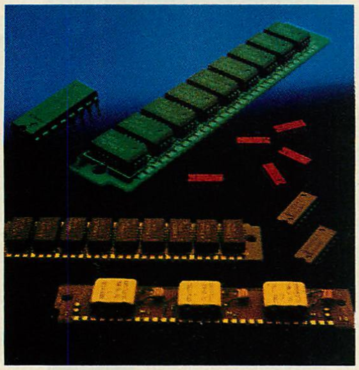
BORLAND
INTERNATIONAL

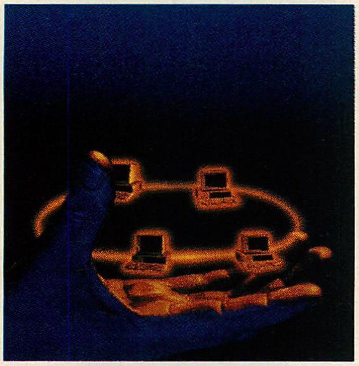
For the dealer nearest you or to order call
(800) 543-7543.

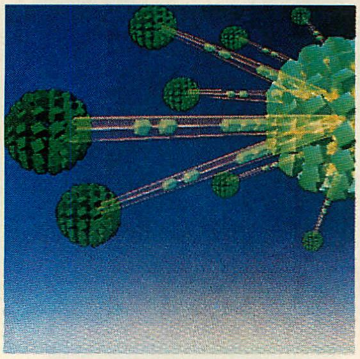
BORLAND

PC TECH JOURNAL®
FOR SYSTEMS DEVELOPERS AND INTEGRATORS









COVER SUITE: STATE OF C

We have watched C grow from a toddler when we first examined it in 1984, through its childhood when we next looked in 1986, to its current state as a young adult. In this issue, Marty Franz reveals a mature language with a field of professional compilers that can only do it justice.

Product review: Nine C compilers

C CONTENDERS
MARTY FRANZ

By most accounts, more PC software developers choose C than any other programming language, leading to a strong and competitive market for C compilers. We limited our field to professional-level compilers: C Ware's DeSmet C; Computer Innovations' C86Plus; Datalight's Optimum-C; Ecosoft's Eco-C88; Lattice's MS-DOS C; Manx's Aztec C86; Mark Williams' Let's C; Metaware's High C; and Microsoft's C 5.0. Earlier versions of all but one of these products (High C) were reviewed in the 1986 installment of the State of C. The only really new developments in the compiler market have been the integrated environments of Borland's Turbo C and Microsoft's QuickC. Their command-line versions are also put through our benchmarks and compared with the other more traditional compilers—with some surprising results. 52

Product reviews: Turbo C QuickC

TURBO AND QUICK WEIGH IN
MARTY FRANZ

The young upstarts deserve separate attention. They offer an integrated programming environment while still maintaining high power and performance—and at the relatively low price of $99. Following in the footsteps of the successful Turbo Pascal and Turbo BASIC, Borland's Turbo C offers a complete compiler in civilized surroundings. Microsoft's QuickC, which comes as part of C 5.0 or can be purchased separately, is equivalent to C 5.0 in its support of the C language—although it lacks a few of the more advanced features. We put Turbo and Quick on the scales to see if either one qualifies as a heavyweight. 72

COMPUTER SYSTEMS

MEMORY IN THE HOT SEAT
STEVEN ARMBRUST and TED FORGERON

Those new 386-based machines running at 16 MHz and higher require faster methods for handling memory. Simple DRAM is no longer effective because it keeps the CPU waiting. Newer memory architectures used in 386 machines today include simple static RAM, interleaved DRAM, page-mode (or static-column) RAM, and memory caching. We explain how each one works and what the trade-offs are in price and performance. 84

VOLUME 6, NUMBER 2

DEPARTMENTS

Cover Illustration · Marc David Cohen

TECH PC JOURNAL

VOL. 6, NO. 2

PUBLISHER: *Donald J. Byrnes*
EDITORIAL DIRECTOR: *Will Fastie*
EDITOR: *Julie Anderson*
EDITORIAL CONSULTANT: *Marjory Spraycar*

EDITORIAL
EXECUTIVE EDITORS: *Susan Holly, David Methvin*
MANAGING EDITOR: *Gail Shaffer*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Maxine Fontana, Philip N. Hisley, Steven S. King, Ted Mirecki*
ASSOCIATE EDITORS: *Douglas Tallman, Jordene Zeimetz*
SENIOR COPY EDITORS: *Bruce Ansley, Maya Brahmam*
COPY EDITOR: *Judith Estrin*
NEW PRODUCTS EDITOR: *Carole Eyring*
LAB TECHNICIAN: *Todd Bannar*
EDITORIAL ASSISTANT: *Barbara Smith*
BUSINESS MANAGER: *Trish Ledbetter*
EDITORIAL SECRETARY: *Beth Sauberlich*
RECEPTIONIST: *Sharleen Sillaman*
CONTRIBUTING EDITORS: *Steven Armbrust, Dave Browning, Peter C. Coffee, Richard M. Foard, Ted Forgeron, Thomas V. Hoffmann, Max Stul Oppenheimer*

ART & PRODUCTION
ART DIRECTOR: *Sharon Reuter*
ASSOCIATE ART DIRECTOR: *Courtney Barone*
ASSISTANT ART DIRECTOR: *Crystal Hopkins*
PRODUCTION MANAGER: *Alison Regan Mrohs*
CONTRIBUTING ARTISTS: *Maciek Albrecht, David Povilaitis*

ADVERTISING SALES
NATIONAL SALES MANAGER: *Rita Burke*
NATIONAL TELEMARKETING MANAGER: *Arlene Braithwaite*
ADVERTISING MANAGER/WEST COAST: *Phyllis Egan*
MARKETING DIRECTOR: *Marti Cunha*
ASSISTANT TO THE PUBLISHER: *Lucy Frazer*
ADVERTISING COORDINATOR: *JeanMarie Donlin*
RESEARCH COORDINATOR: *Kimberly Schroeder*
MARKETING ASSISTANT: *Debbie Lintner*
DISTRICT MANAGERS: *Bill Barney—Southeast and Eastern Canada; Melinda Maginn—New England; John Blake—Mid-Atlantic; Mike Leonard—Midwest; Steve Moorman, Gayl Sorota, Deborah Gisonni, Elaine Ebner—West Coast*
ACCOUNT REPRESENTATIVES: *Mary Martin—Southeast and New England; Nannette Vilushis—Midwest and Mid-Atlantic; Patti Ahlswede, Donna Wood Tyrus—West Coast; Classified Advertising Director—Kathryn Cumberlander*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*
RETAIL SALES MANAGER: *Carol Benedetto*

GROUP VICE PRESIDENT: *J. Scott Briggs*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Co.*
PRESIDENT: *Kenneth H. Koppel*
GROUP VICE PRESIDENTS: *J. Scott Briggs, William Lohse*
SENIOR VICE PRESIDENT, Marketing: *Paul H. Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Research: *Marian O. White*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT: *Hugh Tietjen*
BUSINESS MANAGER: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
CHAIRMAN: *Philip B. Korsant;* PRESIDENT: *Kenneth H. Koppel;* SENIOR VICE PRESIDENT: *Philip Sine;* VICE PRESIDENTS: *Laurence Usdin, William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Selwyn I. Taubman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC Tech Journal, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. FAX (group 3): 301/740-8809. MCIMail: PCTECH. PCTECHline: 301/740-8383. Telex: 6502565932 MCI.

ADVERTISING OFFICES
Corporate/Southeast/Canada: Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044; 301/740-8300. New England: 144 Watchung Avenue, Upper Montclair, NJ 07043; 201/509-8148. Mid-Atlantic: 89 Pearsall Place, Merrick, NY 11566; 516/546-7505. Midwest: Suite 1400, 180 N. Michigan Avenue, Chicago, IL 60601; 312/346-2600. West Coast: 3460 Wilshire Blvd., Los Angeles, CA 90010; 213/387-2100. 110 Marsh Drive, Foster City, CA 94404; 415/378-5580.

SUBSCRIPTION INQUIRIES
PC Tech Journal, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing. One Park Avenue, 4th floor, New York, NY 10016.

PC Tech Journal (ISSN 0738-0194) is published by Ziff-Davis Publishing Co., a division of Ziff Communications Co., One Park Ave., New York, NY 10016. Published monthly except semi-monthly in December. Subscription rate is $34.97 for one year (13 issues). Additional postage for Canada and Foreign is $.60/copy or $8.00/year. Second-class postage paid at New York, NY, and at additional mailing offices. POSTMASTER: Send address changes to PC Tech Journal, P.O. Box 2968, Boulder, CO 80321.

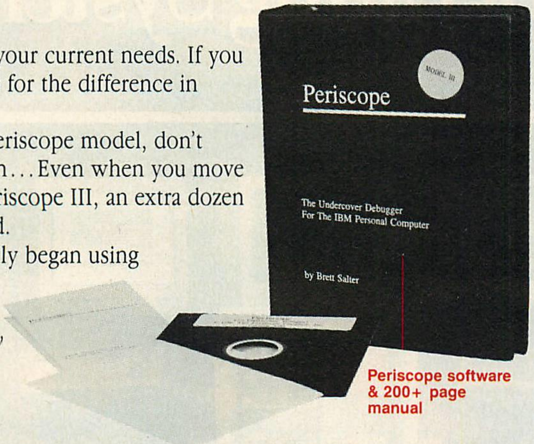
PERISCOPE™ POWER

...Keeps you going full steam ahead when other debuggers let you down! With four models to pick from, you'll find a Periscope that has just the power you need.

Start with the model that fits your current needs. If you need more horsepower, upgrade for the difference in price plus $10!

When you move to another Periscope model, don't worry about having a lot to learn... Even when you move to the most powerful model, Periscope III, an extra dozen commands are all that's involved.

A Periscope I user who recently began using Periscope III writes, *"I like the fact that within the first half hour of use I was debugging my program instead of learning to use the debugger."*

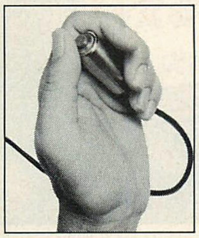


Periscope software & 200+ page manual

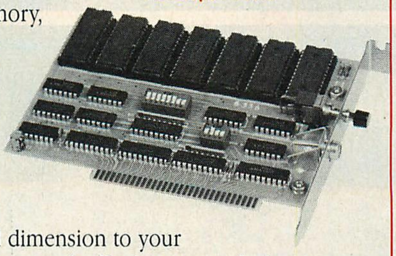
■ **Periscope's software is solid, comprehensive, and flexible.** It helps you debug just about any kind of program you can write... thoroughly and efficiently.

Periscope's the answer for debugging device-drivers, memory-resident, non-DOS, and interrupt-driven programs. Periscope works with any language, and provides source and/or symbol support for programs written in high-level languages and assembler.

■ **Periscope's hardware adds the power to solve the really tough debugging problems.** The break-out switch lets you break into the system any time. You can track down a bug instantly, or just check what's going on, without having to reboot or power down and back up. That's really useful when your system hangs! The switch is included with Periscope I, Periscope II, and Periscope III.

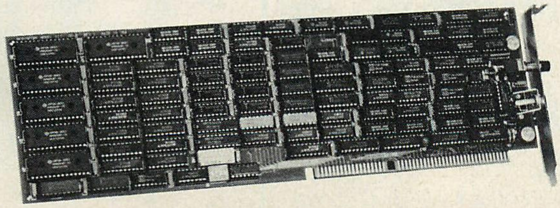

Periscope Break-Out Switch

Periscope I has a board with 56K of write-protected RAM. The Periscope software resides in this memory, safe from run-away programs. DOS memory, where debugger software would normally reside, is thus freed up for your program.


Periscope I Board

Periscope III has a board with 64K of write-protected RAM, which performs the same function as the Periscope I protected memory. AND...

The Periscope III board adds another powerful dimension to your debugging. Its hardware breakpoints and real-time trace buffer let you track down bugs that a software-oriented debugger would take too long to find, or can't find at all!


Periscope III Board

What Periscope Users Like Best:

"I like the clean, solid design and the crash recovery."
Periscope I user

"I like the ability to break out of (a) locked up system!"
Periscope II user

"I am very impressed with Periscope II-X ... it has become my 'heavy duty' debugger of choice, especially if I need to work on a memory resident utility or a device driver."
Periscope II-X user

"... Periscope III is the perfect answer to the debugging needs of anyone involved in real-time programming for the PC ... The real time trace feature has saved me many hours of heartache already."
Periscope III user

- **Periscope I** includes a half-length board with 56K of write-protected RAM; break-out switch; software and manual for $345.
- **Periscope II** includes break-out switch; software and manual for $175.
- **Periscope II-X** includes software and manual (no hardware) for $145.
- **Periscope III** includes a full-length board with 64K of write-protected RAM, hardware breakpoints and real-time trace buffer; break-out switch; software and manual. Periscope III for machines running up to 8 MHz is $995; for machines running up to 10 MHz, $1095.

REQUIREMENTS: IBM PC, XT, AT, or close compatible (Periscope III requires hardware as well as software compatibility); DOS 2.0 or later; 64K available memory; one disk drive; an 80-column monitor.

Call us with your questions. We'll be happy to send you free information or help you decide on the model that best fits your needs.

Order Your Periscope, Toll-Free, Today!
800-722-7006
MAJOR CREDIT CARDS ACCEPTED

The PERISCOPE Company, Inc.

1197 PEACHTREE ST.
PLAZA LEVEL
ATLANTA, GA 30361
404/875-8080

CIRCLE NO. 215 ON READER SERVICE CARD

Call Today and
Receive a FREE Poster
of over 600 Concurrent DOS Applications

Powerful Multiuser Multitasking Concurrent™ DOS 386 Operating System

MAXIMUM PERFORMANCE . . .

Unleash The Power Of The 386!

Concurrent DOS uses the potential of the 386, offering DOS compatibility, combined with multitasking for single-user, and multiuser environments. Up to ten users can share the resources of a single system through simple, easy-to-connect, serial terminals as configured . More users can be added by purchase of a Concurrent DOS System Builder's Kit. Multiuser, Multitasking Applications can be designed using the Concurrent DOS Programmer's Toolkit.

MAXIMUM COMPATIBILITY . . .

Runs Concurrent DOS Multitasking Applications And All The Popular PC DOS Applications.

The applications running on your current system (Lotus® 1-2-3®, dBase® III, WordPerfect® and many more) are still useable and don't have to be replaced with "work-alikes" or "compatibles". Concurrent DOS 386 contains windowing capabilities allowing up to four of these applications to run from the primary system console. Concurrent DOS will also allow popular DOS applications to be run from serial terminals.

CIRCLE NO. 221 ON READER SERVICE CARD

DIGITAL RESEARCH

Concurrent DOS APPLICATIONS

DATA BASE

Ansa Software
PARADOX

Data Access Corporation
*DATA FLEX

Qne International
Q PRO 4

Unlimited Processing
*TEAM-UP

DENTAL

Articulate Publications
DENTALIS-DENTAL OFFICE PACKAGE

Data Modes, Inc.
DENTAL PATIENT ACCOUNTING

Digital Software Corporation
PAS-3 PLUS-DENTAL BILLING SYSTEM

Professional Economics Bur... Inc.
ASYDENT

Raynbow Technolog...
DENTIQUE

Systems Plus, Inc.
DDS-MEDICAL MANAGER

DESKTOP PUBLISHING

Xerox Corporation
XEROX VENTURA PUBLISHER EDITION
XEROX DOCUMENTER

EDUCATION/SCHOOLS

Barden & Associates, Inc.
TRAINING PAC

W.L. Slaughter & Associates
SCHOOL MANAGEMENT SYSTEM

The Williamson Group
MICRO-ALUM/ALUMNI DEVELOPMENT SYSTEM
MICRO-ADMIT/ADMISSIONS OFFICE SYSTEM
APPLY/COLLEGE GUIDANCE OFFICE SYSTEM

Data M...
INVEN...

Digital ...
GOLD ...
INVEN...

Hardware ...

GOLD ACCO...
COSTING

Mc Software, I...
*INCOME II-BIL...
*INCOME II-JO... PROGRESS
*INCOME II-MAT... REQUIREMENT...
*INMASS-JOB CO... PROGRESS

Micro Management ...
JOB COST

Realworld Corporation
JOB COST
LABORATORY MANAGE...

Genlex, Inc.
GEN LAB-TEST LABORATORY

ON SPEC REMOTE LINK
ON SPEC REMOTE SUPPORT
FOXBORO SPEC 200 MICRO/ON SPEC

Information Products, Inc.
INVENTORY MANUFACTURING

MC... Inc.
...GRATED MANUFACTURING
...MATERIALS
REQUIREMENTS

Call 1-800-443-4200 for a complete up-to-date listing.

...GER)-

CBR Softw...
MEDICAL SY...

Data Modes, Inc.
PATIENT ACCOUN...

Dataforms, Inc.
AUTOMATED MATERIALS MANAGEMENT

Digital Software Corporation, Inc.
PAS-3 PLUS-MEDICAL BILLING SYSTEM

Micro Resources
MICRO RESOURCES SYSTEM MENUS
MICRO RESOURCES "WYSE" PAC

New Generation Systems, Inc.
...EE
...SPEED BACK-UP/RESTORE
...-COMPUPRO UTILITY PACKAGE
...ENT DOS PROGRAMMER
...OLS
...LL

...MPILER

...ING C
...EN DEBUGGER
...RO

...KS

...RECTORY
... PRINTER

...ARCH CBASIC ...ILER™ (CB-86)
DIGITAL RESEARCH PASCAL/MT +™
...GITAL RESEARCH PL/1™
...AL RESEARCH FORTRAN 77™
...L RESEARCH ASSEMBLER PLUS

... Planning Corporation
...BASIC

Innovation Computer Systems
MARSHAL PASCAL

Manx Software Systems, Inc.
*AZTEC-C

Minnow Bear Computer
CBS TOOLS (CB-86)
CBS MANAGER (CB-86)
CBS XREF (CB-86)
MB PASLIB (PASCAL MT +)
MB + TOOLS (PASCAL MT +)

O'Hanlon, Inc.
DATA BASE SOLUTION
CLASSIC SOLUTION

PROJECT MANAGEMENT

MINIMUM INVESTMENT . . .

Migrate to the Multiuser World With Minimum Investment!

Protect your development investment with easy migration within the Intel® microprocessor family. Plus, a library of Concurrent DOS multiuser applications already exists to meet the diverse requirements of many user environments. From medical practices to manufacturing floors, the price/performance capabilities of Concurrent DOS 386 are readily available.

Minimize your investment in software, training, and support, while you take advantage of the latest hardware/software/system technologies with Concurrent DOS 386. Unleash Concurrent DOS 386 and discover the operating system that puts power at your fingertips!

SPECIAL FEATURES

- PC DOS 3.3 Compatible
- Full support of all models of IBM® Personal System/2™ Series.
- True Multitasking With Simultaneous Execution
- Supports Multiusers Sharing Built-In Networkable Resources
- User-Friendly
- Easy to Install and Use
- Up to 255 Simultaneous Tasks
- Support for Large Memory Systems
- Over 1000 Business Solutions Available Worldwide
- Full Complement of Development Tools Available
- Three Times Faster Screen I/O Than MS® -DOS
- Serial Port Configurability Up to 38.4K Baud

DIGITAL RESEARCH®

We've spent years developing our file manager so you won't have to.

You could invest hundreds of programming hours writing a file management system for your next application.

Or you could simply invest in Btrieve.® And get all the file handling functionality you need, through subroutine calls from your favorite programming language.

Portable. Write your application once. Wherever Btrieve runs, your application will run, whether in a single user or multiuser environment. In fact, Btrieve is the standard access to NetWare.®

Fast. Written in assembly language, Btrieve uses b-tree indexing algorithms with caching and automatic balancing for fast, efficient file management.

Safe. Btrieve is the only fault tolerant file manager with built-in file recovery. In the event of a system or power failure, your database is protected.

Flexible. Develop applications with the capabilities you need most. Like 255 open files, unlimited records per file, 24 indexes per file, and a maximum file size of up to four gigabytes. You can access Btrieve from BASIC, C, Pascal, COBOL and others.

Invest in Btrieve. At just $245 for single user and $595 for multiuser, it's a small price to pay for all the file manager you'll ever need.* And you'll never pay royalties on the applications you develop. To find out more, see your authorized Novell reseller, or call (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVBT3.

NOVELL

For software solutions, you should be seeing red.

Novell European Headquarters, Novell GMBH, Schiess-Strasse 55, 4000 Düsseldorf 11, West Germany
Tel. Nat. (0211) 5973-0, Tel. Int. 49 211 5973-0, FAX 49 211 59 11 49, TELEX 85887

CIRCLE NO. 201 ON READER SERVICE CARD

JULIE ANDERSON

SYSTEMS PERSPECTIVE

JAnderson

C Comes of Age

Our ongoing efforts to measure the progress of C show that the language is reaching maturity.

We have come a long way from the premiere issue of *PC Tech Journal* in which author William Hunt listed criteria for evaluating C compilers ("How to Choose a C Compiler," July/August 1983, p. 78). The points we were concerned about then should make us smile now: "You should be able to edit, compile, and link C programs without swapping floppy disks," Hunt wrote, and "Compiling and linking small files should take less than two minutes."

In the third and fourth issues of our young magazine, Hunt applied his criteria to nine C compilers ranging in price from $100 to $550 ("C and the PC," Part 1, November/December 1983, p. 110 and Part 2, January 1984, p. 91). At that time, the compilers were distinguished mainly by their level of compliance with the Kernighan and Ritchie specifications for the C language and standard C library. Reflective of the limited memory available on early PCs, most compilers supported only the small memory model (64KB of code, 64KB data). Compile times were respectable for a two-diskette PC. As for performance of compiled code, Hunt concluded, "None produces the kind of very compact and highly optimized programs needed for a large software product that strains the PC's resources."

Two years later Hunt subjected 12 compilers to an updated set of criteria ("The State of C," January 1986, p. 82). This time he found that the market was segmented along two lines: compilers for personal use priced from $49.95 to $109, and tools for professional development ranging from $350 to $500.

All compilers supported the standard C library, and all professional compilers supported the full C language. The professional compilers also supported five memory models: large, medium, compact, small, and tiny. Add-on utilities made compilers distinct as vendors offered librarians, assemblers, linkers, and source-level debuggers. The beginnings of a development environment grew up around the previously isolated compiler.

Another two years have passed and the time has come again to take the pulse of C. In this month's cover suite beginning on page 52, Marty Franz presents the 1988 installment on the state of C. He reviews the full range of professional compilers. Among the new developments since the last review is ANSI's involvement; it has prepared a draft standard for extensions to the C language and library. Most compilers have embraced these needed extensions, even at this proposal stage.

Perhaps the two most significant developments in the past year have been Borland International's entry into the C market with Turbo C and Microsoft's rapid response, QuickC. These compilers are reviewed separately as integrated environments in "Turbo and Quick Weigh in" (p. 72).

Modifying the successful formulas of Turbo Pascal and Turbo BASIC, Borland provides a complete professional compiler enveloped in an integrated environment—all for $99. The only missing element is a debugger, promised for the first quarter of 1988.

Microsoft's approach is quite different. QuickC is marketed two ways: bundled (as a prototyping tool) with Microsoft C 5.0 or by itself. QuickC and C 5.0 are functionally equivalent in their support of C and share a common library. The main difference is that the C 5.0 optimizing compiler produces the best performing code of all compilers we reviewed.

An integrated environment is at a disadvantage because the tools provided with the environment are often not as powerful as the editor and debugger that the developer can choose separately. QuickC's integrated debugger, for example, supports only the medium memory model.

However, the overriding advantages to integrated development environments are the close coupling of data throughout the environment and the intelligent communications among the parts. The compiler works with the editor to pinpoint source-code errors, and the debugger traces the source-code lines as each is executed. Integrated environments are attractive for repetitive edit, compile, and debug loops, but having a powerful optimizing compiler ready to polish the code completes the ideal development cycle.

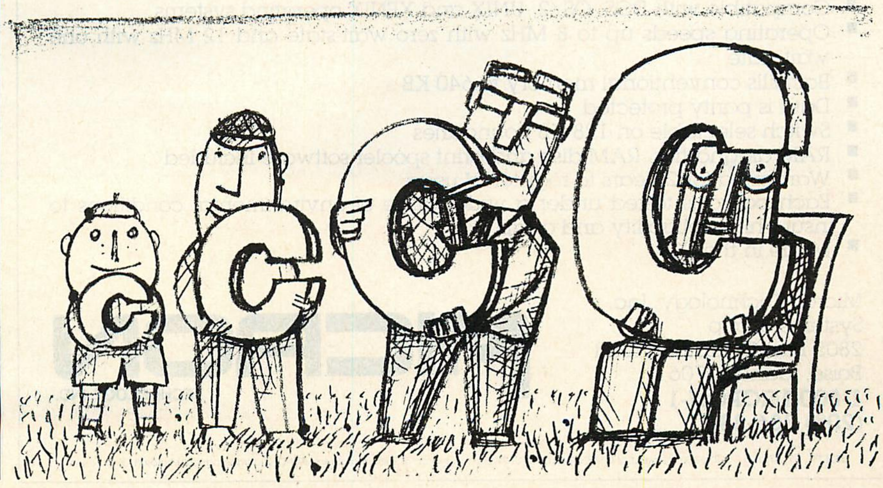


ILLUSTRATION • MACIEK ALBRECHT

All the compilers we reviewed here are designed for DOS development. As we went to press, we received IBM's C/2 and MASM/2 for developing programs to run under OS/2. This new generation of development tools will be the subject of an upcoming article.

NEW FACES, NEW BYLINES

PC Tech Journal has changed at least as much as the state of C since 1983, the year the magazine was born. When we were first getting started, the combined years of systems experience of our technical staff was well under 25. As I write this, to announce additions to both our technical and publishing staffs, our combined years of systems experience is 105—all of which helps us to understand what you as systems developers and integrators need to know in order to do your job.

Joining us in November 1987 as a technical editor was Steven S. King, a specialist in distributed systems who can count the planning and installation of more than 20 local area network systems to his credit. He will coordinate our LAN coverage.

The second addition to the technical staff is Philip Hisley, who comes to us with more than 21 years of systems development experience specializing in the design and implementation of real-time process control systems. In his role as technical editor, Hisley will be directing our coverage of the C, Pascal, and Ada languages, UNIX, developers' tools, and graphics systems.

Associate editor Jordene Zeimetz joins us with seven years of experience as a reporter, writer, and editor for technical publications. In April she will inaugurate a new reader response column in which she will report on the replies to our reader's opinion cards. (You will find one in front of this editorial. Please fill it in and send us your opinion on the usefulness of the C language.) Zeimetz has a bachelor's degree in political science and journalism from the University of Maryland.

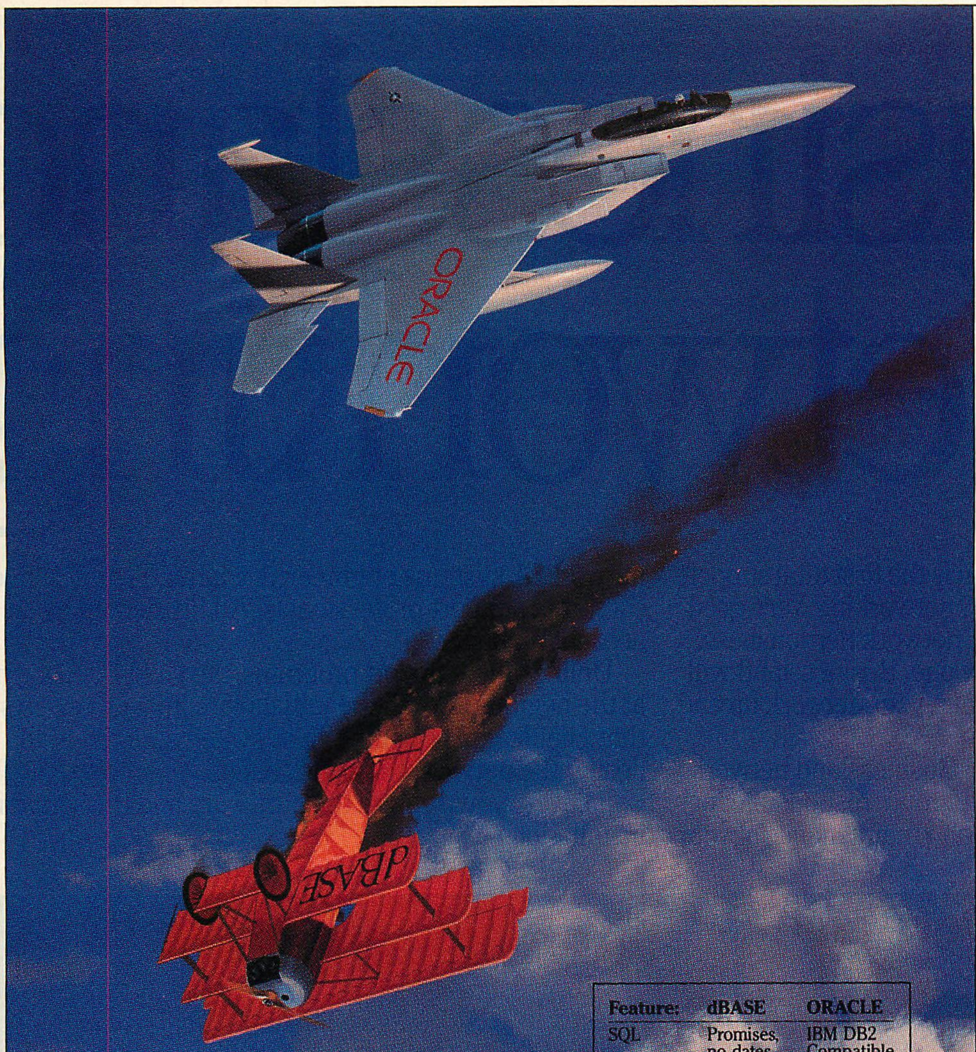
Our second associate editor, Douglas Tallman, has been a newspaper reporter for seven years and was honored by the Maryland, Delaware, D.C. Press Association with an award for one of his columns. Tallman, who earned a bachelor's degree in journalism from the University of Rhode Island, is the Macintosh software librarian for a local Apple computer user's group.

A new addition to our copy edit staff is Judith Estrin, who has several years experience in publishing technical material. She has a bachelor's degree in English and Biology and a master's in publications design.

The last new face is lab technician Todd Bannar. Although fresh out of school, his computer-related part-time job experience rivals some seasoned professionals. He has responsibility for supervision of our lab, the administration of our local area network, and the management of our electronic program listing service, PCTECHline.

Bannar's first order of business was to install an improved version of PCTECHline. More telephone lines have been added to provide easier access. The telephone number remains 301/740-8383 with communication parameters of 300, 1,200, or 2,400 bps, no parity, 8 data bits, and 1 stop bit.

You might say that we, like C, have come of age. We have assembled a staff of talented journalists, writers, and editors, and we are poised to deliver relevant and helpful information that you need to face the next new age.



Feature:	dBASE	ORACLE
SQL	Promises, no dates	IBM DB2 Compatible
Mainframes	No Way	IBM MVS & VM/CMS
Minis	Nope	DEC, HP, Sun, etc.
PCs	All, PC Jr. too	286 & 386 PCs
MS/DOS	<640K programs	>640K programs
OS/2	Ask Ashton-Tate	Yes, first day
Multiuser	Primitive	Mainframe quality
Networking	PC Nets only	PC, mini & mainframe
Fault Tolerant	You must be kidding	CPU & Disk Recovery

THE LAST DBMS
ONLY $199
CALL 1-800-ORACLE1

Oracle Corporation, the world's fastest growing software company,[1] has just climbed past Ashton-Tate to become the world's largest supplier of database management software and services.[2]

Why?

• Because ORACLE® runs on PCs, plus mainframes and minicomputers from IBM, DEC, DG, HP, Prime, Wang, Apollo, Sun, etc.— virtually every computer you have now or ever will have. Ashton-Tate's dBASE runs only on PCs.

• Because ORACLE is a true distributed DBMS that connects all your computers—PCs, minicomputers and mainframes—into a single, unified computing and information resource. dBASE supports only primitive PC networking.

• Because Oracle has supported the industry standard SQL language since 1979. Ashton-Tate promises to put SQL into dBASE sometime in the indefinite future.

• Because ORACLE takes advantage of modern 286/386 PCs by letting you build larger-than-640K PC applications on MS/DOS today, and run them unchanged on OS/2, once OS/2 is available. dBASE treats today's 286/386 PCs and PS/2s like the now obsolete, original PC.

Don't go down in flames. Bail out from dBASE. Call 1-800-ORACLE1 and order your $199-PC copy of ORACLE[3] today. Or just ask and we'll send you information on ORACLE, the number one selling DBMS on minicomputers and mainframes.

ORACLE®

COMPATIBILITY • PORTABILITY • CONNECTABILITY

Call 1-800-ORACLE1, ext. 148 today.

Dear Oracle,

PC ORDER PROCESSING
Oracle Corporation
20 Davis Drive • Belmont, CA 94002

I want ORACLE to be THE LAST DBMS for my 286/386 PC. Enclosed is my ☐ **Check** or ☐ **VISA** ☐ **MC** ☐ **AMEX** credit card authorization for $199 (California residents add 7% sales tax). I understand this copy is for PC development only. Offer valid only in the US and Canada.

Print Name ______ Date ______

Title ______

Company ______

Street (*P.O. Box numbers not acceptable*) ______

City ______

State ______ Zip ______

Phone ______

Credit Card Number ______

Card Expiration Date ______

Signature ______ PCTECHJOURNAL

I am a value-added reseller (VAR): ☐ YES ☐ NO

[1] Revenue doubled in 9 of Oracle's 10 years. [2] Sales rate over $200 million in current fiscal year. [3] For PC development use only. Requires a 286/386 PC plus 1-MByte extended memory. Offer valid only in US & Canada. TRBA

Businessland Networking

The Businessland® Gold Plan is the first total support program to ensure successful PC networking. Very good news if you're experiencing pre-network nervousness.

It's a guarantee that, from day one, Businessland will be there with you. Helping to keep your network up and productive.

The heart of this plan is your Businessland network support team. One experienced group, responsible for the design, installation, service and maintenance of your network. A systems engineer. A trainer. Technical and corporate support professionals. All headed up by an account manager.

Unlike other support programs, the Gold Plan begins at the beginning. With the initial planning and reviews that enable your Businessland systems engineer to design the network to fit your needs.

It covers you through the hardware and the software installation.

Introduces With A Net.

As well as file maintenance, user set-up, menus, application installation, delivery and system tests.

It means we'll be there to train your network administrator, and assures ongoing post-sale consulting, on-site reviews and continuing support. Plus instant access to our technical experts via an 800 phone line.

There's even a unique service that lets us access your network by phone for remote diagnosis. And, in many cases, instant repair.

If you're preparing to make your networking decision, talk to Businessland and find out more about the Gold Plan.

Call (800) 323-1000. We'll tell you how to contact the Businessland nearest you for a free consultation.

BUSINESSLAND®

A Different Kind of Computer Company

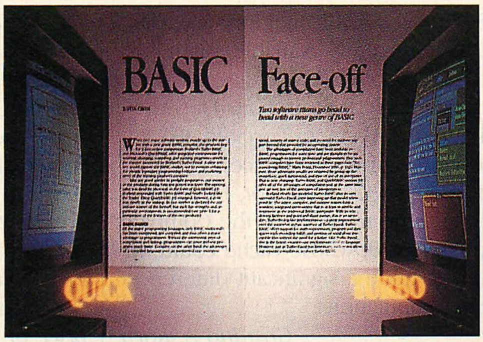
Cathy Heumann
Connectivity Instructor

Kevin Compton
Corporate Technical Support Manager

Larry Cromwell
Systems Engineer

CIRCLE NO. 140 ON READER SERVICE CARD

LETTERS



NO BASIC LIMITATIONS

Justin Crom's comparison of QuickBASIC and Turbo BASIC ("Basic Face-Off," September 1987, p. 136) in *PC Tech Journal* was a much needed, thorough, and insightful comparison of two excellent products. His examination of the strengths and weaknesses of these fine products should allow an informed decision for anyone who is trying to decide between them. However, I found a couple of errors in his evaluation of QuickBASIC.

Under the subheading, BASICA Compatibility, Mr. Crom asserts that "QuickBASIC and BASICA allow up to 255 characters [in their source-code lines]." In fact, QuickBASIC allows virtually unlimited source-code line length by use of the underscore and carriage return at least every 255 characters. I have several literal strings that are 1,040 and 2,000 characters long and have used command lines up to 450 characters in length.

In table 2 of the article, QuickBASIC is portrayed as having no method to set the size of the stack. In fact, the stack can be set using the CLEAR ,,*stack* command, where *stack* is any value between 512 and the amount of memory left in the DS: segment. With a program consisting solely of the statement CLEAR ,,*stack* this value is given as 59,286.

Except for the aforementioned errors, Mr. Crom's comparison/evaluation was right on target. Now, how about a comparison using QuickBASIC 4.0?

R. Stephen Whiteaker
30th Medical Group
APO NY

Mr. Whiteaker is correct in that Microsoft's QuickBASIC 3.0 uses the underscore as a continuation character, thereby allowing the creation of "logical" lines in excess of the 255 character limitation of the physical line. I apologize for overlooking this fact.

Table 2 in the article lists only the options for controlling the program's environment at compile time, whereas QuickBASIC's CLEAR statement is a source-language statement that takes effect at run time. Although it can reset the stack size, CLEAR also sets all numeric variables to zero, all strings to null, and closes all files opened by the BASIC program. Its effect, therefore, depends on its placement in the source code. QuickBASIC has no position-independent option equivalent to Turbo BASIC's $STACK metastatement.

A review of QuickBASIC 4.0 is in progress. One difference already discovered has to do with Mr. Whiteaker's first point, continuing long lines. Details will be published in a future issue of PC Tech Journal.

—Justin Crom

A MISTAKEN CONNECTION?

I applaud your efforts to include articles on mainframe connectivity in your journal. I found Michael Hurwicz's article ("Connectivity Pathways: APPC or NETBIOS," November 1987, p. 156) to be quite interesting and informative. There are several points made, however, that I believe are misleading as well as incorrect.

Mr. Hurwicz states, "APPC is a particular type of LU [logical unit] known as LU6.2." This is correct; furthermore, IBM uses APPC and (Systems Network Architecture) LU6.2 interchangeably. He then goes on to list IBM's Enhanced Connectivity Facility (ECF) as "other APPC implementations" in the sidebar. This implies that the ECF is an APPC/LU6.2 implementation, which it is not. In fact, the ECF is a rather limited-function, micro-to-mainframe communications solution that does *not* use LU6.2. The ECF, for its part, is implemented on a 3270 terminal emulation session using LU2.

In his discussion of ECF, the author states, "SRPI [Server-Requester Programming Interface] will be available, initially at least, only for PCs and System/370s, not for System/3x machines." IBM has offered S/3x functions equivalent to those offered by the SRPI (virtual disk, virtual print, upload/download) for several years. The product is called PC Support/3x. The PC Support/36 product runs via the terminal emulation session (not LU6.2) between the PC and the S/36. Using the product, it is possible to do the following: use the S/36 printers as PC printers (Virtual Print), use S/36 disk as PC disk shared among several users (Virtual Disk/Shared Folders), and upload/download data automatically and selectively by using an SQL-like interface (Source Transfer Facility).

Kevin W. Tolly
BMW of North America, Inc.
Montvale, NJ

You are correct that ECF is currently implemented using LU2, not LU6.2/APPC. However, IBM has stated that ECF can be used as a migration path to APPC: a future implementation of ECF will be implemented using APPC as the protocol. Although the services of PC Support/3x provide similar functionality to those of the ECF, they are not compatible products.

—DM

LEADING THE PACK

I enjoyed your review of the PC's Limited 386[16] ("Price/Performance Leader," Susan Holly and Jim Shields, December 1987, p. 150), with its excellent performance table. You correctly pointed out the lack of 80387 math coprocessor support. I believe that, for your many readers now making purchase decisions, this point is dramatically important, moreso than most other design features. This applies not just to engineering-type number crunchers, but to anyone who uses a large spreadsheet. A 386 clone without 80387 support is,

quite simply, not a cost-effective choice, because fast 16-bit AT clones do very nearly as well at a much lower price. By contrast, even the cheapest wait-state-infested 386 clone equipped with a 387 coprocessor can run circles around an 80287-equipped machine in any floating-point computation.

The crucial point is this. The 386 makes only 16-bit memory accesses with most currently available software; this picture will improve. The 287 can make *only* 16-bit memory accesses. But the 387 makes 32-bit memory accesses, even with currently available, plain-vanilla, DOS software!

Incidentally, also in your December issue, the Compaq Portable 386 ("The Power of Convenience," David Claiborne, p. 132) does *not*, I believe, have the same memory as the Compaq Deskpro 386/20, whose cache could, in fact, make quite a significant difference in real computations.

Granino A. Korn
Tucson, AZ

For those users for whom math coprocessor support is a serious concern, the extra benefit provided by the 80387 compared with an 80287 is worth the extra cost. The purchaser of a 386-based computer who either is in or plans to be in this category should be careful to purchase a machine that supports the 387. A substantial number of people, however, may wish to buy a 386-based machine (particularly at the price of a PC's Limited 386[16]) solely for its fast CPU and its enhanced ability to support the simultaneous execution of the DOS applications.

You are correct that the Compaq Portable 386 and Deskpro 386/20 have different memory architectures. As noted in our December review, the Portable 386 has a paged memory architecture like that used on the original Compaq Deskpro 386. The Deskpro 386/20 uses a cache memory architecture that provides better performance in most situations. For additional information on these two memory architectures, see "Memory in the Hot Seat," Steve Armbrust and Ted Forgeron, this issue (p. 84). We will be publishing a complete review of the Deskpro 386/20 in an upcoming issue.

—SH and JS

OBJECTIVELY YOURS

The 17-page advertisement for the IBM System 9370 that you printed in the November 1987 edition ("The 9370 Attraction," Dennis Linnell, p. 174) under the guise of objective journalism was irresponsible. It removes all possibility that your publication will ever be taken seriously by myself or any of my clients as an independent, objective vehicle for computing information.

First and foremost, an article describing a computer that starts at $70,000 is out of place for a PC magazine. If you are going to start evaluating this class of machine, please do not forget the other major vendors in this market space: Data General, Digital, Hewlett Packard, Unisys, etc.

I hope that in the future you will live up to the disclaimer on your masthead page that states the following (and I quote): "*PC Tech Journal* is an independent journal, not affiliated in any way with International Business Machines Corporation."

Christopher Williams, president
C. L. Williams Company
Perrysburg, OH

Our masthead is correct. We are not affiliated in any way with IBM. How-

ever, when IBM, the largest computer manufacturer in the world, releases a product designed to be connected to IBM-standard PCs, we take note.

As a magazine for systems developers and integrators, PC Tech Journal's *job is to provide solutions to connectivity problems. The 9370 is one option. However, we realize that IBM is not the only vendor offering solutions, so we also will be looking at significant computers from other vendors in the context of their "connectability" to PCs.*

—JA

C—POINTS OF VIEW

The letter from Wilson Jones ("Doesn't C," Letters, November 1987, p. 9) put into words what I have been thinking for years with regard to the utility of BASIC compared to other languages, especially low-level ones.

My primary job responsibility has been sales and marketing, but I have had to write many business application programs in BASIC to support these other functions. Even though I have been a part-time and self-taught programmer, I believe that the hundreds of hours spent in this area has given me a certain level of programming expertise and understanding.

I have always wondered what was so mysteriously great about other languages being touted by many computer professionals. I read that FORTRAN was the language of science and engineering, COBOL was the language of business, Pascal was an emerging global standard, and so on. My programming experience has covered a wide range of applications, including science and engineering, and I have yet to find a specific problem that cannot be handled in BASIC.

With the affordable compilers now available, BASIC currently has near-assembler speed. And, like Mr. Jones, I cannot understand why the computing world isn't stampeding to standardize around such a simple, universal, and portable programming language.

J.R. McLemore
Calabrian Chemicals Corporation
Houston, TX

I'd like to respond to the letter written by Wilson Jones and published in *PC Tech Journal.* I am both surprised and concerned that one who has programmed for 20 years doesn't recognize the benefits of coding in a language as powerful, flexible, and portable as C. (Of course, the letter could have been a joke.)

C does have the ability to perform almost as a Super Macro Assembler: it provides a "bit-twiddling" ability. It also provides commands that are available only in higher-level languages. That means that C is flexible—a programmer can program as efficiently or as sloppily as desired, and someone can come back and tighten up the code if necessary. This is especially true in those on-line dialogues when the user waits for that subsecond response time. I know some of my users would have some choice (possibly legal) words for me if I told them to go buy a bigger/faster machine because I couldn't code to their machine and requirements.

The C language promotes structured code. I've had nightmares of having to maintain some 20-year-old COBOL program that is chock full of ALTERs and GOTOs.

C is portable. It still baffles me that someone would write an application in BASIC for an IBM microcomputer. Then, if the same application were needed for an AT&T 3B, the programmer would have to turn around and write it in another language. Later, if this application were needed for a big-iron IBM mainframe, it would have to be rewritten a third time.

Granted the development costs could always be passed on to the user. Or if the user wants it in COBOL, then the user gets it in COBOL. But I think that the same user will get tired of paying for such development costs when other packages are available in C that could run on all three with only a recompile. Ashton-Tate has dBASE running on PCs and AT&T 3Bs. Microsoft has Word doing the same. Informix is doing it too. With IBM's announcement of its intention to develop a C compiler for mainframes, how much development effort will it be to get either of these products up and running on a mainframe? This is assuming that IBM doesn't stray too far away from what is standard C on the micros and minis.

Just think . . . a pork-belly commodity tracking program could work equally well and with minimal effort for the brokerage's mainframe, the butcher shop's minicomputer, and the pig farm's PC. What would you rather do? Code it only *once* in C, or code it once in BASIC, once in RPG, and once again in COBOL?

There *is* one drawback to C . . . It doesn't keep your coding skills honed in other languages.

Robert D. Wilson
Plano, TX

GETTING INTO SYNCH

I would like to bring your attention to a few erroneous statements in the article by J. Scott Haugdahl ("DOS-LAN Juncture," July 1987, p. 78).

In this article, the author refers to a scenario about sharing files in compatibility mode that cannot occur. He implies that it is possible for two users to share a file but yet get out-of-synch when user B updates a record just read by user A. The implication herein is that user A would not be aware that he/she is using out-of-date information because DOS performs local cache buffering on the files that are opened in compatibility mode.

The reason why such a scenario is incorrect is that the *only* circumstances under which two or more processes can successfully open a file initially opened in compatibility mode (using system call 0x3DH) is if the file is *read-only* and if any subsequent requests open this file in compatibility mode with read access specified or in deny-write or deny-none mode with read access specified.

Since the attribute of the file is read-only, this implies that any process attempting to write to a file opened in the modes described above, will encounter an access-denied message when this operation is attempted.

In addition, figure 4 of the article specifies that when two users access the same file in compatibility mode, regardless of the access mode specified, the result of the second and subsequent opens to this file are successful. This is not correct. If the attribute of the file is read/write, subsequent requests to open this file will fail. If the file is read-only, this file can be opened by more than one process *only* when read access is specified.

With the exception of the statements previously noted, this was an extremely informative article and obviously a well-researched one.

Stephan C. Moen
Taylor Management Systems
Rosemont, IL

You are correct that buffering problems within DOS do not occur when two different processes are using compatibility mode. It is possible, however, for a single process to open a file multiple times for read/write access in compatibility mode. In that situation, buffers maintained by the process (in the application's code) could cause the synchronization problems described.

As you point out, the file-sharing matrix in figure 4 is incorrect for multiple opens in compatibility mode. If the open fails, INT 24H (the critical error handler) will be called, signalling a sharing violation. Thank you for your correction and clarification.

—DM

ORIGIN OF THE B-TREE

In the evaluation of Btrieve/Xtrieve by Burks A. Smith ("A Data Manager with Language Flexibility," October 1987, p. 104) in *PC Tech Journal*, the article states that Btrieve uses the binary-tree indexing scheme. In fact, Btrieve does not utilize a binary-tree data structure, but instead uses a multiway tree definition of a B-TREE. The B-tree structure was developed by R. Bayer and E. McCreight at the Boeing Scientific Research Labs in the early 1970s and is one of the most popular methods that are available for organizing index structures. A binary-tree structure allows only two children at any level, whereas a B-tree data structure is endowed with the following properties: (1) A B-tree node of order M has between $1/2M$ and M subtrees at every node, except at the root and leaf levels; (2) the root of a B-tree has at least two subtrees,

NODE WARRIORS.

For many people, using their Local Area Network is like a journey across 500 miles of bad desert, with strange, unforeseen forces of destruction lurking behind every turn.

Until now that is.

Because now there's TOPS, a revolutionary departure in LAN design. TOPS fully arms each node with local user control of information, and frees you from the grips of costly central file servers—and even more costly network administration personnel.

TOPS' unique design gives users completely transparent access to files anywhere from any operating system on any computer—anywhere in the network. And TOPS' modular design supports new operating systems as they become available. Users become believers, because TOPS is so easy to operate. All its commands are totally intuitive.

TOPS is also easy to install and support, and there's no downtime necessary when adding new users. You just plug them in. And because there's no central file server you avoid data crashes and bottlenecks. Your LAN will be stronger than ever, and faster too, because users communicate directly with each other.

Reinforcements!

TOPS Version 2.0 does everything the original does, then adds a few extras all its own. Like automatic publishing and mounting of folders and volumes. Access to locally attached PC printers by other networked PC users. Compatibility with Apple Filing Protocol, and FlashTalk™ speed (it's three times faster than AppleTalk in PC to PC communications). Translation of PC files to Mac format and vice versa. And a lot more.

And price? No need to worry, because it won't put you into deficit spending. TOPS is just $189 per node for Macintoshes and PCs.

So call 800-222-TOPS (US), 800-445-TOPS (CA), for a free Node Warrior button and more information. It'll tip the balance of power in your direction.

TOPS®
A Sun Microsystems Company
2560 Ninth Street, Berkeley, CA 94710

Registered trademark: TOPS, Sun Microsystems, Inc.
Trademarks: FlashTalk, Sun Microsystems, Inc; AppleTalk, Macintosh, Apple Computer, Inc.

CIRCLE NO. 127 ON READER SERVICE CARD

unless it is a leaf; (3) all leaves of the tree are at the same level.

I hope that the above explanation is helpful in distinguishing between B-trees and binary trees.

John S. Koperwas
Exhibition Place
Ontario, Canada

We are embarrassed because we certainly know the differences between B-trees and binary trees. We even ran a two-part article describing B-trees, where we pointed out these differences ("Tree Structures," Atindra Chaturvedi, February 1985, p. 78; "Tree Structures, Part 2," March 1985, p. 131). We make every effort to weed out such misstatements—but this one got past us.

—JA

OS/2 COMPATIBILITY BOX

With reference to Will Fastie's sidebar ("Compatibility Coffin," New Directions, November 1987, p. 21), I have the following comment: I suggested to IBM two years ago that the compatibility box should be an 80x86 coprocessor card supported by an OS/2 driver. In retrospect, that idea looks better (and cheaper and faster) than ever to me.

Dave Feustel
Austin, TX

ERRATA

In "Succeeding C" (Marty Franz, September 1987, p. 166), in table 1 on page 177, the minimum RAM requirement for Guidelines C++ should correctly read 640KB.

In the "HTEST/HFORMAT" Product Watch in the December issue (Peter Aitken, p. 197), the vendor of WordPerfect was cited incorrectly. The correct vendor is, of course, WordPerfect Corporation. *PC Tech Journal* sincerely regrets these errors.

COMMENTS WELCOME

All letters to the editor should be directed to Editor, *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. Correspondence also can be submitted over MCI Mail to PCTECH.

Although *PC Tech Journal* cannot publish all letters received, every effort is made to answer as many as possible. Please keep letters brief and to the point, and include name, mailing address, and telephone number; when a letter is lengthy, a diskette is appreciated.

Upgrade your technology

The software technology available to programmers of IBM-compatible personal computers is truly amazing. And newer, more powerful development packages appear all the time. But until now, finding out about these important products has been a difficult and time consuming task.

FREE Buyer's Guide. The New Programmer's Connection Buyers Guide contains individual descriptions of over 500 titles of programmer's development software by over 150 manufacturers. Each description covers major product features as well as any software or hardware requirements and version numbers. In the box on the right are some examples of the types of descriptions you'll find in our Buyer's Guide.

No Hidden Charges. The low discount prices in our Buyer's Guide are all you pay. We don't charge extra for domestic UPS Ground shipping, credit cards, COD orders, purchase orders, sales tax (except Ohio) or special handling (except for non-Canadian international orders).

Guarantees. We offer FREE 30-day no-risk return guarantees and 30-day evaluation periods on most of our products.

Latest Versions. The products we carry are the latest versions and come with the same manufacturer's technical support as if buying direct.

Large Inventory. We have one of the largest inventories of programmer's development products in the industry. Most orders are shipped within 24 hours.

Noncommissioned Staff. Our courteous salespeople are always ready to help you. And if you aren't sure about your needs, our knowledgeable technical people can give you sound, objective advice.

Experience. We've specialized in development software for IBM-compatible personal computers since 1984 and are experienced at providing a full range of quality products and customer services.

How to Get Your Copy. There are three ways for you to receive your FREE copy of the Programmer's Connection Buyer's Guide: 1) Use the reader service card provided by this journal; 2) Mail us a card or letter with your name and address; or 3) Call one of our convenient toll free telephone numbers.

If you haven't yet received your Programmer's Connection Buyer's Guide, act now. Upgrading your programming technology could be one of the wisest and most profitable decisions you'll ever make.

Copia International

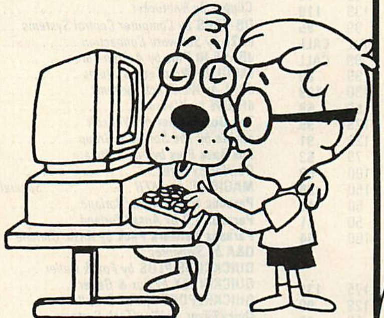
Peabody

Pop-up Resident Reference Utility

List $100 Reg $89
Special Introductory Price $79
through 02/28/88

Peabody is a fast and flexible on-line reference utility with databases for Microsoft C, Turbo Pascal and MS-DOS. Peabody spares the manual and spoils the programmer by providing instant, accurate and complete information in pop-up frames at the touch of a key. Whether you need a simple reminder or a full description and useful examples, you'll find it with Peabody. Each database has been designed and assembled by seasoned programmers; their applied experience is obvious in the concise and practical information that unfolds at your request in overlapping frames. Peabody gives you two ways of finding that information; select general topics from a structured subject menu or use Peabody's unique Hyper-Key to get instant help for the keyword closest to the cursor. Other special features that make Peabody an outstanding value are its ability to reference multiple databases with a single keystroke; its Sticky Frame feature, which lets you paste information on screen for reference when you return to your editor; and display functions which list on screen the contents of disk files and computer memory in text or binary formats (ASCII or EBCDIC). Peabody can run as a standalone program, in tandem with your text editor, or as a memory-resident utility. The Peabody disk includes a setup utility for customizing Peabody and a compiler for integrating your own subject categories, keywords and reference material directly into a Peabody database.

Requires as little as 85K memory. **Comes with a full unconditional 60-day money-back guarantee.**



CALL TOLL FREE

USA:	**800-336-1166**
Canada:	**800-225-1166**
Ohio & Alaska (Collect): ..	**216-494-3781**
International:	**216-494-3781**
Telex:	**9102406879**
Easylink:	**62806530**

Programmer's Connection
7249 Whipple Avenue NW
North Canton, OH 44720

Please turn the page for our latest price list and ordering information. →

PROGRAMMER'S QUALITY INTEGRITY Service SUPPORT PRICE CONNECTION

The Free Programmer's Connection Buyer's Guide.

CIRCLE NO. 175 ON READER SERVICE CARD

ai - expert systems

	List	Ours
1st-CLASS *by Programs in Motion*	495	399
EXSYS *Development Software by EXSYS*	395	289
LEVEL5 *by Information Builders*	685	569
Logic-Line Series *All varieties by Thunderstone*	CALL	CALL
RuleMaster 2 *by Radian Corporation* *New*	CALL	CALL
VP-EXPERT *by Paperback Software*	125	79

ai - lisp language

muLISP-87 Interpreter *by Soft Warehouse*	300	199
muLISP-87 Interpreter & Compiler	400	259
Q'Nial *Various by NIAL Systems*	CALL	CALL
Star Sapphire LISP *Compiler by Sapiens*	495	429

ai - prolog language

Arity Combination Package	1095	979
Expert System Development Pkg	295	229
File Interchange Toolkit	50	44
PROLOG Compiler & Interpreter	650	569
Screen Design Toolkit	50	44
SQL Development Package	295	229
Arity PROLOG Interpreter	295	229
Arity Standard Prolog	95	77
LPA microPROLOG *All Varieties*	CALL	CALL
MPROLOG P550 *w/Primer by LOGICWARE*	220	159
Turbo PROLOG *by Borland Intl*	100	64
Turbo PROLOG Toolbox *by Borland Intl*	100	64

ai - smalltalk language

Smalltalk/V	100	84
EGA/VGA Color Option	50	45
Goodies Diskette #1	50	45
Smalltalk/Comm	50	45

ai - texas instruments

Arborist *Decision Tree Software*	595	519
PC Scheme *Lisp*	95	77
Personal Consultant Easy	495	435
Personal Consultant Images	495	435
Personal Consultant Online	995	869
Personal Consultant Plus	2950	2589
Personal Consultant Runtime	95	84

ada language

AdaVantage *GSA-validated by Meridian Software*	795	735
AdaVantage Utility Packages	50	47
DOS Environment Package	50	47
Ada *GSA-validated w/maintenance by alsys* *New*	3355	3119
AdaQUERY *New*	200	185
Ada Developer's Toolset *Volumes 1 & 2* *New*	995	919

apl language

APL*PLUS *by STSC, Specify PC or PS/2*	695	495
APL*PLUS PC Spreadsheet Mgr *by STSC*	195	139
APL Tools *by STSC*	295	199
ATLAS*GRAPHICS *by STSC*	450	329
Financial/Statistical Library *by STSC*	275	189
Pocket APL *by STSC*	95	69
STATGRAPHICS *by STSC*	895	649

assembly language

386/ASM/386 LINK *Cross Asm by Phar Lap*	495	389
ASMLIB *Function Library by BCSoft*	149	125
asmTREE *B-Tree Dev System by BCSoft*	395	329
Cross Assemblers *Various by 2500 AD*	CALL	CALL
DMS Resident-ASM *by American Software Intl.* *New*	150	139
Microsoft Macro Assembler	150	CALL
OPTASM *by SLR Systems* *New*	195	CALL
risC *by IMSI* *New*	80	69
Turbo Debugger *by Speedware*	89	79
Turbo Editasm *by Speedware*	99	84
Visible Computer: 8088 *by Software Masters*	80	64

basic language

ApBasic *by CompTech* *New*	99	79
db/Lib *for QuickBASIC by AJS Publishing*	139	119
Finally *by Komputerwerk*	99	85
MACH 2 *by MicroHelp*	CALL	CALL
Microsoft QuickBASIC	99	CALL
QBase *Relational Database by Crescent*	99	89
Quick-TOOLS *by BCSoft*	130	109
QuickPak *by Crescent Software*	69	59
Scientific Subroutine Library *by Wiley*	125	99
Screen Sculptor *by Software Bottling*	125	91
Stay-Res *by MicroHelp*	79	53
True BASIC	100	69
True BASIC *w/Run-time*	150	99
True BASIC 3D Graphics	50	41
True BASIC Developer's Toolkit	50	41
Turbo BASIC Compiler *by Borland Intl*	100	64

blaise products

ASYNCH MANAGER *Specify C or Pascal*	175	135
C TOOLS PLUS/5.0	129	99
KeyPlayer *Super Batch Program*	50	45
LIGHT TOOLS *for Datalight C*	100	65
PASCAL TOOLS/TOOLS 2	175	135
RUNOFF *Text Formatter*	50	45
Turbo ASYNCH PLUS/4.0	129	99
Turbo C TOOLS	129	99
Turbo POWER TOOLS PLUS/4.0	129	99
VIEW MANAGER *Specify C or Pascal*	275	199

borland products

EUREKA *Equation Solver*	167	105
Quattro: The Professional Spreadsheet *New*	195	125
Reflex: The Analyst	150	99
Sidekick	85	57
Superkey	100	64
Turbo Basic Compiler	100	64
Turbo Basic Database Toolbox	100	64
Turbo Basic Editor Toolbox	100	64
Turbo Basic Telecom Toolbox	100	64
Turbo C Compiler *(Call for support products)*	100	64
Turbo Lightning and Word Wizard	150	94
Turbo Lightning	100	64
Turbo Lightning Word Wizard	70	47
Turbo Pascal	100	64
Turbo Pascal Database Toolbox	100	64
Turbo Pascal Developer's Toolkit	395	289
Turbo Pascal Editor Toolbox	100	64
Turbo Pascal Gameworks Toolbox	100	64
Turbo Pascal Graphix Toolbox	100	64
Turbo Pascal Numerical Methods Toolbox	100	64
Turbo Pascal Tutor	70	41
Turbo Prolog Compiler	100	64
Turbo Prolog Toolbox	100	64

c language

C-terp *by Gimpel, Specify compiler*	298	219
C Trainer *with Book by Catalytix*	123	87
DC88 *by C Ware* *New*	99	89
DC88 *with Large Case Option by C Ware* *New*	178	159
Eco-C88 Modeling Compiler *by Ecosoft*	100	79
Instant C *by Rational Systems*	495	369
Lattice C Compiler *vers. 3.2 from Lattice*	500	265
Mark Williams Let's C *with FREE csd*	75	54
Microsoft C Compiler 5.0 w/CodeView	450	CALL
Microsoft QuickC Compiler	99	CALL
Optimum-C *by Datalight*	139	95
Turbo C Compiler *by Borland*	100	64
Turbo C-terp *for Turbo C by Gimpel* *New*	139	119
Uniware 68000/10/20 Cross Compiler *by SDS*	995	829

c utilities

Blackstar C Library *by Sterling Castle*	125	98
C++ *by Guidelines* *New Version*	295	259
c-tree & r-tree Combo *by FairCom*	650	519
c-tree *ISAM File Manager*	395	315
r-tree *Report Generator*	295	239
Curses Window Dev Pkg *by Aspen Scientific*	119	105
with Source Code	289	249
dBx *dBASE to C Translator by Desktop AI*	350	299
with Source Code	550	419
DMS Resident-C *by American Software Intl* *New*	150	139
Entelekon Combo Package *Special Price*	200	99
Flash-up *by Software Bottling*	89	78
GraphiC *by Scientific Endeavors* *New Version*	395	309
HALO *Graphics by Media Cybernetics*	300	205
HALO Development Pkg *for Microsoft*	595	389
The HAMMER *by OES Systems*	195	129
Heap I/O *by System Software* *New*	CALL	CALL
HOOPS *3-D Graphics by Ithaca Software* *New*	575	459
LINK & LOCATE *by Systems & Software* *New*	350	309
PANEL *by Roundhill, Specify QuickC or Turbo C*	129	95
PANEL Plus *by Roundhill*	495	395
PC Lint *by Gimpel Software*	139	99
RTC PLUS *Fortran to C by Cobalt Blue*	450	369
Sapiens V8 *Virtual Memory Manager*	300	265
Scientific Subroutine Library *by Wiley*	175	135
TE Developer's Kit *by Sub Systems*	95	85
Vitamin C *by Creative Programming*	225	149
VC Screen *Forms Designer*	100	79
WKS LIBRARY *by Tenon Software* *New*	89	79
Zview *by Data Mgmt Consultants*	245	119

cobol language

COBOL spII *by Flexus*	395	329
Micro Focus COBOL *See Micro Focus Section*		
Microsoft COBOL *See Microsoft Section*		
Realia COBOL with RealMENU	1145	899
Realia COBOL	995	783
RealCICS	995	783
RM/COBOL *by Ryan-McFarland*	950	CALL
RM/COBOL 85 *by Ryan-McFarland*	1250	CALL
RM/USER+5 *by Ryan-McFarland*	250	CALL
RM/Screens	395	CALL
SCREENIO *by Norcom*	400	379

database management

Advanced DBMaster *by Macon Software*	510	419
Advanced Revelation *by COSMOS* *New*	950	CALL
Clipper *by Nantucket* *New*	695	379
DB-FABS *by Computer Control Systems* *New*	295	CALL
dB2c *by Software Connection* *New*	299	CALL
dBASE III Plus *by Ashton-Tate*	695	389
dbSQL *by WordTech Systems*	CALL	CALL
dBXL *by WordTech Systems*	169	109
dFLOW *by Wallsoft*	149	119
The Documenter *by Wallsoft*	295	225
enable *by The Software Group* *New*	CALL	CALL
Fox Base Plus *by Fox Software* *New*	395	249
Genifer *by Bytel*	395	275
MAGIC PC *by AKER* *Special Price,New*	199	179
Paradox 1.1 *by Ansa/Borland*	495	359
Paradox 2.0 *by Ansa/Borland*	725	525
Paradox Network Pack *by Ansa/Borland*	995	725
Q&A *by Symantec*	349	219
QUICKCODE PLUS *by Fox & Geller*	295	169
QUICKINDEX *by Fox & Geller*	149	95
QUICKREPORT *by Fox & Geller*	295	169
QuickSilver *by WordTech Systems*	599	349
R:Base 5000 *by Microrim*	495	CALL
R:Base System V *by Microrim*	700	CALL
\rdb *by Robinson-Shafer-Wright*	139	119
SQL Base *by Gupta Technologies* *New*	995	CALL
Multi-User Version *New*	1995	CALL
Tom Rettig's Library *by Tom Rettig & Assoc*	100	79
UI Programmer *by Wallsoft*	295	239
VP-INFO *by Paperback Software*	125	79
VP-PLANNER *by Paperback Software* *New*	100	63
VP-PLANNER PLUS *by Paperback Software* *New*	125	79
XDB II *by Software Systems Technology* *New*	395	CALL
C Programming Interface *New*	295	CALL

debuggers & profilers

386 DEBUG *Cross Debugger by Phar Lap*	195	129
Advanced Trace-86 *by Morgan Computing*	175	115
Codesmith-86 *by Visual Age*	145	98
DSD87 *by Soft Advances*	125	79
MiniProbe *by Atron*	395	369
Periscope I *with Board by Periscope*	345	275
Periscope II *with NMI Breakout Switch*	175	139
Periscope II-X *Software only*	145	105
Periscope III *8 MHz version*	995	795
Periscope III *10 MHz version*	1095	875
The PROFILER *with Source Code by DWB*	125	89
TURBOsmith *Source debugger for Turbo Pascal*	99	69
The WATCHER *Profiler by Stony Brook*	60	51

disk utilities

Back-It *by Gazelle Systems*	130	115
Disk Optimizer *by Softlogic Systems*	60	55
Disk Technician *by Prime Solutions*	100	89
FASTBACK *by 5th Generation Systems*	159	115
FASTBACK PLUS *by 5th Generation Systems* *New*	189	139
Take Two Manager *United Software Security*	139	119
Vcache *by Golden Bow Systems*	50	47
Vopt *by Golden Bow Systems*	50	47
Vfeature Deluxe *by Golden Bow Systems*	120	111
XenoCopy-PC *by XenoSoft*	80	69

dos utilities

Advanced Norton Utilities	150	89
Command Plus *by ESP Software*	80	69
Desqview *from Quarterdeck*	130	115
FANSI-CONSOLE *by Hersey Micro*	75	62
Mace Utilities *Paul Mace Software*	99	89
MicroHelp Utility *by MicroHelp*	59	49
Norton Commander *by Peter Norton*	75	55
Norton Utilities *by Peter Norton*	100	59
Q-DOS II *by Gazelle Systems*	70	59
Taskview *by Sunny Hill Software*	80	55
The Weiner Shell *by Gryphon Microproducts* *New*	199	CALL
XO-SHELL *by Wyte Corporation* *New*	69	CALL

essential products

Breakout Debugger *by Essential Software*	125	89
C Utility Library	185	118
Essential Communications	185	118
Essential Communications *with Break Out*	250	189
Essential Graphics	250	183
/*residentC*/	99	85
with Source Code	198	148
ScreenStar	99	85
with Library Source Code	198	154

forth language

FORTH/83 Metacompiler *Specify Target*	750	599
MMS Forth *by Miller Microcomputer Svcs* *New*	180	159
PC/FORTH *by Laboratory Microsystems*	150	109
PC/FORTH+ *by Laboratory Microsystems*	250	188
Programmer's Package #1 *by LMI*	250	199
Programmer's Package #2 *by LMI*	350	279
Programmer's Package #3 *by LMI*	500	399
UR/FORTH *Also Available for OS/2 by LMI*	350	258
UR/FORTH Libraries	500	395

fortran language

50 MORE: FORTRAN *by Wiley*	125	95
ACS Time Series *by Alpha Computer Service*	495	389
AUTOMATED PROGRAMMER *by KGK Automated*	995	899
Essential Graphics *by Essential Software*	250	183
Forlib-Plus *by Alpha Computer Service*	70	44
FORTRAN Addenda *by Impulse Engr*	165	138
HALO *Graphics by Media Cybernetics*	300	205
I/O PRO w/No Limit Library *by MEF*	250	219
Microcompatibles Combo Package	240	215
Grafmatic	135	117
Plotmatic	135	117
Microsoft FORTRAN *w/CodeView*	450	CALL
No Limit Library *by MEF Environmental*	129	109
Numerical Analyst *by MAGUS*	295	199
PANEL *by Roundhill Computer Systems*	295	199
RM/FORTRAN *by Ryan-McFarland*	595	399
Scientific Subroutine Library *by Wiley*	175	135
Statistician *by Alpha Computer Service*	295	235
Strings & Things *by Alpha Computer Service*	70	45

greenleaf products

Greenleaf C Sampler *specify QuickC or Turbo C*	95	69
Greenleaf Comm Library	185	125
Greenleaf Data Windows Library	225	155
with Source Code	395	249
Greenleaf Functions	185	125

help utilities

HELP/Control *by MDS*	125	99
On-line Help *from Opt-Tech*	149	99
SoftScreen/HELP *by Dialectic Systems*	195	149

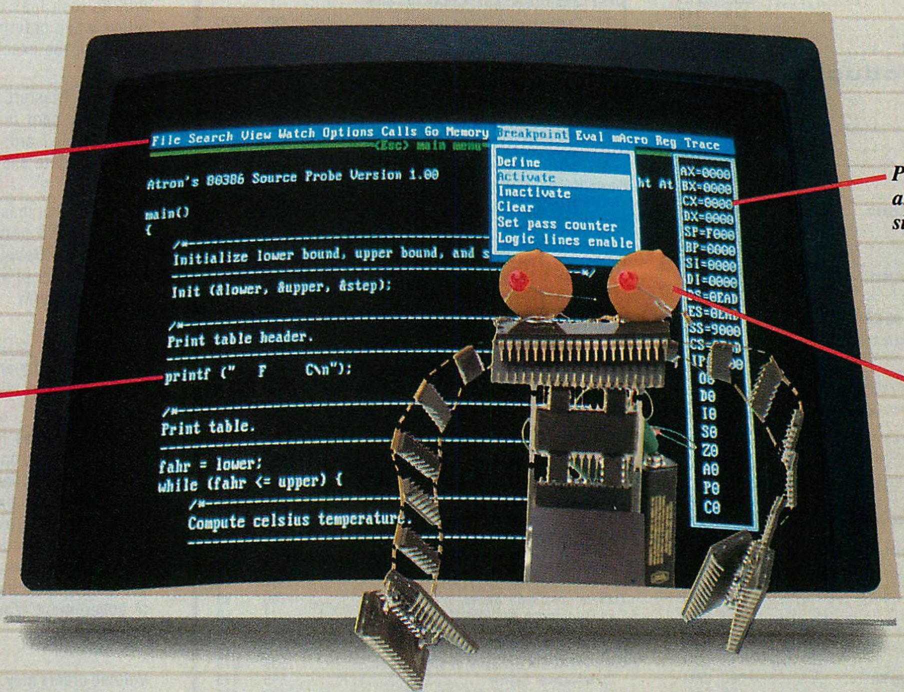
lattice products

Lattice C Compiler *ver 3.2 from Lattice*	500	265
with Library Source Code	900	495
C Cross Reference Generator	50	37
C-Food Smorgasbord *Function Library*	150	95
with Source Code	300	179
C-Sprite *Source Level Debugger*	175	CALL
Curses *Screen Manager*	125	85
with Source Code	250	169
dBC III	250	169
with Source Code	500	356
dBC III Plus	750	594
with Source Code	1500	1184
LMK *Make Facility*	195	138
RPG II Combo *All four items below* *New Version*	1400	1099
RPG II Compiler *No Royalties* *New Version*	750	625
Screen Design Aid Utility *for RPG II*	350	309
SEU *Source Entry Utility*	250	199
Sort/Merge	250	199
SecretDisk II *Encryption Utility*	79	59
SideTalk *Resident Communications*	120	88
SSP/PC *Scientific Subroutine Library*	350	269
Text Management Utilities	120	88

metagraphics products

FontWINDOW	95	79
LightWINDOW/C *for Datalight C*	95	79
MetaWINDOW *No Royalties*	195	159
MetaWINDOW/PLUS	275	229
TurboWINDOW/C *for Turbo C*	95	79
TurboWINDOW/Pascal *for Turbo Pascal*	95	79

IT'S TIME TO DO SOME SERIOUS 386 BUGBUSTING!



PROBE's menu bar and pull-down menus set a new standard for debugger interfaces.

PROBE has source-level debugging to let you "C" your program.

POP registers up and down with a single key.

This is an out-of-range memory-overwrite bug. Since it is interrupt related, it only appears in real time.

Welcome to your nightmare. Your company has bet the farm on your product. Your demonstration wowed the operating committee, and beta shipments were out on time. Then wham!

All your beta customers seemed to call on the same day. "Your software is doing some really bizarre things," they say. Your credibility is at stake. Your profits are at stake. Your sanity is at stake.

THIS BUG'S FOR YOU

You rack your brain, trying to figure something out. Is it a random memory overwrite? Or worse, an overwrite to a stack-based local variable? Is it sequence dependent? Or worse, randomly caused by interrupts? Overwritten code? Undocumented "features" in the software you're linking to? And to top it off, your program is too big. The software debugger, your program and it's symbol table can't fit into memory at the same time. Opening a bicycle shop suddenly isn't such a bad idea.

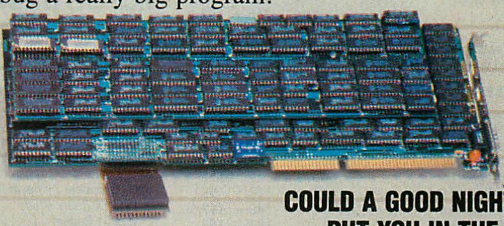
THIS DEBUGGER'S FOR YOU

Announcing the 386 PROBE™ Bugbuster,*from Atron. Nine of the top-ten software developers sleep better at night because of Atron hardware-assisted debuggers. Because they can set real-time breakpoints which instantly detect memory reads and writes.

Now, with the 386 PROBE, you have the capability to set a *qualified breakpoint,* so the breakpoint triggers only if the events are coming from the wrong procedures. So you don't have to be halted by breakpoints from legitimate areas. You can even detect obscure, sequence-dependent problems by stopping a breakpoint only after a specific chain of events has occurred in a specific order.

Then, so you can look at the cause of the problem, the 386 PROBE automatically stores the last 2K cycles of program execution. Although other debuggers may *try* to do the same thing, Atron is the only company in the world to dequeue the pipelined trace data so you can easily understand it.

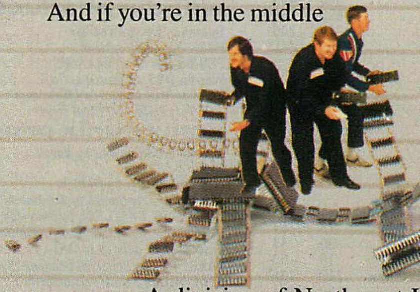
Finally, 386 PROBE's megabyte of hidden, write-protected memory stores your symbol table and debugger. So your bug can't roach the debugger. And so you have room enough to debug a really big program.



COULD A GOOD NIGHT'S SLEEP PUT YOU IN THE TOP TEN?

Look at it this way. Nine of the top-ten software products in any given category were created by Atron customers. Maybe their *edge* is – a good night's sleep.

Call and get your free, 56-page bugbusting bible today. And if you're in the middle of a nightmare right now, give us a purchase order number. We'll FEDEX you a sweet dream.

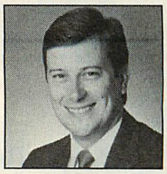


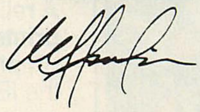
Atron

BUGBUSTERS

A division of Northwest Instrument Systems, Inc.
20665 Fourth Street • Saratoga, CA 95070
408/741-5900

NEW DIRECTIONS





The Trouble with C

C may be the leading software development environment, but it can't be because we love it. □ *And, finally, the real thing: IBM delivers the OS/2 package as promised.*

Several years ago, my good friend, business associate, and long-time *PC Tech Journal* contributing editor Richard Foard wrote an article about a language that was just then coming into its own on the PC. He did it somewhat spontaneously after completing a research project, from which he concluded that C was close to madness as a programming language. He called the article, which he never published, "The Trouble with C."

I liked the concept and asked him to develop his article for *PC Tech Journal*. By the time I asked the question, however, Rich was too busy developing some complex systems using, as you may have guessed, that terrible C. He no longer really felt that C was troubled, so he declined my offer.

He has graciously allowed me to use the title today, for, as you will see, I think it still holds true, and I think you do, too. You may find this surprising, coming as it does in an issue of *PC Tech Journal* that contains our update on "The State of C." You may also find it surprising because our market research shows C to be the most popular language among our readers.

C CONTRADICTIONS

Our recently completed language study found that almost 25 percent of our audience choose C as their primary programming language and almost 20 percent choose it as a secondary language. What that says to me is that 75 percent, a significant majority of our readers, are using something else. The language study points to BASIC (!) as the second most popular language with 17 percent, followed by Pascal at 14 percent. dBASE ranks fourth at 10 percent, while COBOL and FORTRAN make respectable showings of 9 and 7 percent, respectively.

BASIC and Pascal make strong showings as the secondary languages (18 and 15 percent), although assembly language ties with BASIC at 18 percent—not a surprise. FORTRAN comes in at 8 percent, dBASE at 7 percent, and COBOL at 4 percent.

Your reasons for choosing a language are the same, regardless whether the language is for primary or secondary use. Speed of the executable code ranks first. Given the complexity and variety of work you are doing, that is hardly surprising and entirely expected. Perhaps more meaningful, though, are your next three most important characteristics of programming languages: structured code, separate compilation, and string handling. After that, for primary languages, came the availability of I/O libraries, with speed of compilation following close behind; these two qualities were reversed in the case of secondary languages.

These are significant findings because, while speed of the resulting product is most important, the traditional facilities associated with high-level languages are right up there as well. They rank higher than ANSI compliance, integrated editor, symbolic debugger, portability, price, and many others. I am astonished that the integrated environment get lower marks in light of the great interest in Borland's Turbo and Microsoft's Quick products. To be fair, I should point out that many of the results were neck and neck, but I emphasize again that the top four criteria were the same for both primary and secondary languages.

The language study, in conjunction with an earlier *PC Tech Journal* survey of data management software, provides important clues about what you want in a language. We asked, in both studies, what kinds of applications were being developed, regardless of the particular language. The vast majority are financial and accounting, or other business-related software. In the data management study, more than 90 percent of the programs being developed were business applications. In the language study, 46 percent were data management programs, 40 percent were financial applications, and 45 percent were other business-related programs (multiple responses were allowed).

There is one more result—an extremely important one for my argument here. In the data management survey, we asked whether you use data management software (that is, the language provided with a data manager) to develop applications, either in-house or for a client company. Sixty-one per-

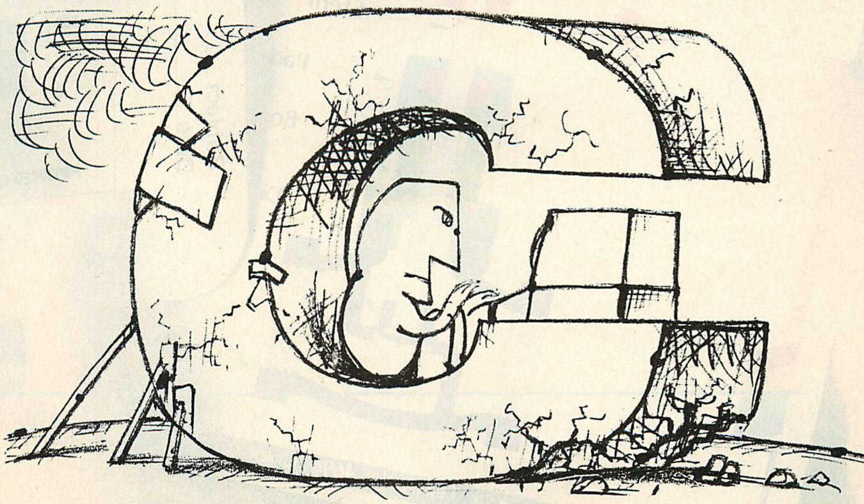


ILLUSTRATION • MACIEK ALBRECHT

TEK COLOR PRINTERS

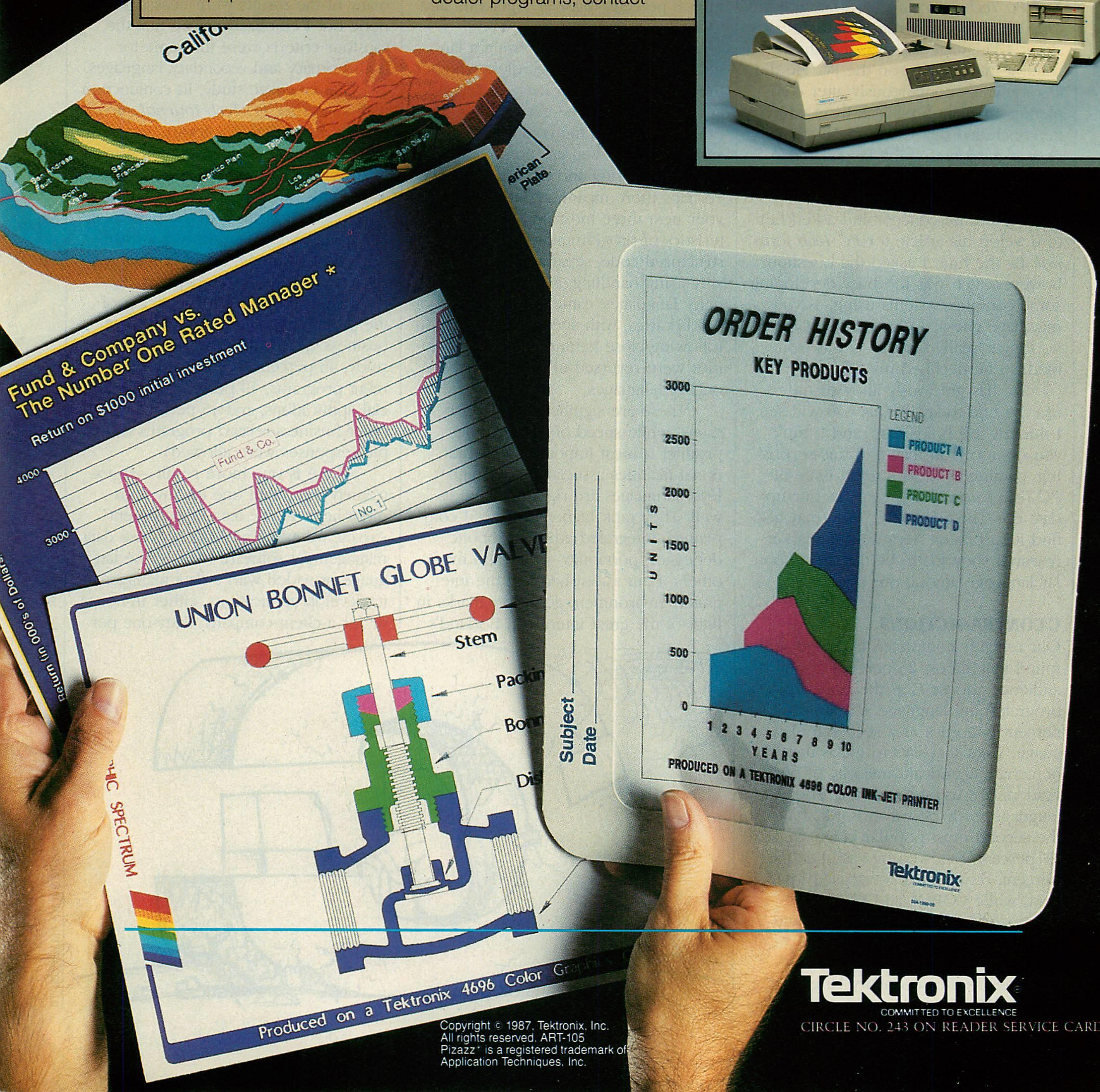
HOW TO MAKE YOUR PC APPLICATIONS LOOK GOOD ON PAPER.

The Tek 4696 Color Printer: A reliable, cost-effective printer. From presentation graphics to solids modeling, the IBM-PC compatible 4696 creates rich, vibrant color on paper or transparencies.

Behind the 4696 is Tek's world-renowned graphics technology and applications expertise. Technology that matches dot size with ink and paper formulation for excellent results.

To enable superb color output from nearly any PC application, you receive a copy of Pizazz® software with your 4696. You also get the full backing of the 67-country strong Tektronix service and support network.

For more information about the complete 4690 Color Printer Family and OEM/VAR dealer programs, contact your local Tek representative. Or call 1-800-225-5434. In Oregon, 1-235-7202.



Tektronix
COMMITTED TO EXCELLENCE

CIRCLE NO. 243 ON READER SERVICE CARD

cent of you said yes for in-house work, and a whopping 74 percent replied yes for client companies.

These results make it absolutely clear that, although C is touted as the number one language, it is not used by the majority of our audience for the majority of applications. Furthermore, conventional languages are much less frequently used than data management software for developing applications. Question: If C is so great, why doesn't it have a larger share of the market?

My answer, and I think it is your answer too, is that C is *not* so great, *not* the most productive computer language tool. In many respects, C is not even a high-level language; it is really more of a high-level macro assembler, one that removes the programmer from the excruciating details of specific machine instructions, registers, flags, and addressing modes but still leaves the programmer very close to the underlying hardware. This is great for systems-level software, like compilers and operating systems, but it is a far cry from the kind of tool that a programmer needs to write straightforward business applications.

C is syntactically cryptic and hard to learn and use. It is therefore difficult to read and maintain C program code. The C language is notorious for allowing the programmer great freedom by limiting various kinds of checks (for example, loose type checking), so it is equally notorious for letting the programmer paint himself into a corner. Debugging C is particularly difficult because most complex C programs are rife with pointers; pointing chains can be so long and complex that it is almost impossible to find the target value or indeed to understand what the type of the target value is.

High-level languages should unload as much programming detail from the programmer as possible. The C language unloads very little.

Of course, high performance or great flexibility is sometimes essential to producing the desired program. That is why some programs still get written in assembly language, and there is nothing wrong with that.

TIME IS MONEY

The professional systems developer, however, has a far more pressing problem: time. Software development is expensive, so whenever performance is not the driving force behind an application, the most important considerations are how long it will take to write and how much the effort will cost. In these instances, data management and object-oriented languages may have the upper hand. It is no mere accident that the percentage of applications that are written using data management products is higher for client companies than it is for in-house deployment; consultants and resellers know that their profit margin depends on expeditious completion of the job.

One of the characteristics of data management languages, and of more traditional-in-style but contemporary-in-implementation languages such as Barrington Systems' Clarion, is that they tend to have numerous built-in facilities for dealing with data (probably the most important) and human interaction (probably the most useful). These facilities leverage the programmer by reducing complex programming chores by ordinary standards to relatively easy and quick tasks. These facilities are especially useful when they eliminate procedural code and require only that the programmer develop descriptions for the tasks to be performed.

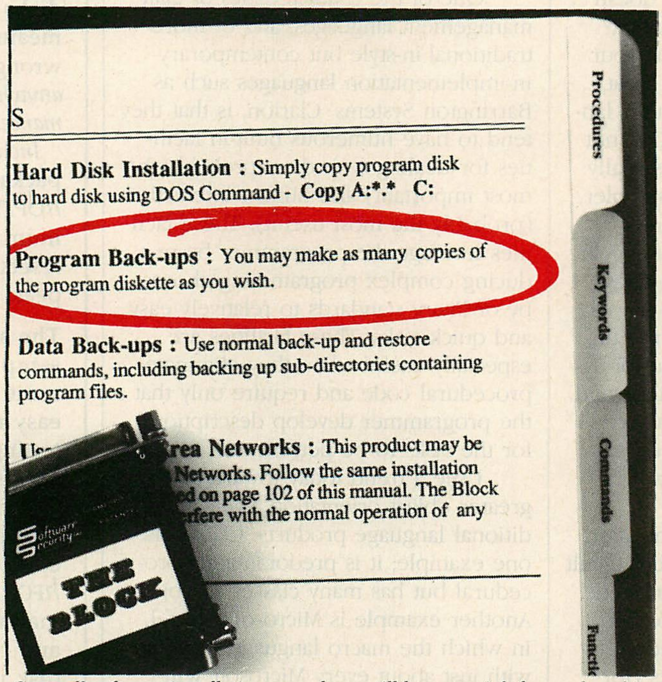
I see a trend today toward a greater object orientation, even in traditional language products. Clarion is one example; it is predominantly procedural but has many classes of objects. Another example is Microsoft's Excel, in which the macro language can deal with just about every Microsoft Windows object as well as many traditional spreadsheet objects.

The biggest objection to such products (aside from emotional ones like "real programmers don't fill out forms") is usually that their performance is lacking, and this is often true. However, the better products in this category do not have the perceived performance problems at all. Sure, benchmarking Borland's Paradox with the Sieve program will yield poor results, but they are misleading because most data management applications are not compute-intensive, reflecting instead a greater balance between computation, manipulation, and I/O.

What the better products do is to optimize the important functions. Paradox provides two excellent examples. The first is the TEXT function, which is used to display multiple lines of text. TEXT is dramatically faster than using either ?? or MESSAGE to display the lines of text one line at a time. The second is the Paradox query function, which is entirely nonprocedural. A Paradox query frees the programmer from almost all syntactic concerns and is

"How to protect your software by letting people copy it."

By Dick Erett, President of Software Security



Soon all software installation procedures will be as straightforward as this. The only difference will be whether you include the option to steal your product or not.

Inventor and entrepreneur, Dick Erett, explains his company's view on the protection of intellectual property.

"A crucial point that even sophisticated software development companies and the trade press seem to be missing or ignoring is this:

Software protection must be understood to be a distinctively different concept from that commonly referred to as copy protection.

Fundamentally, software protection involves devising a method that prevents unauthorized use of a program, without restricting a legitimate user from making any number of additional copies or preventing program operation via hard disk or LANs.

Logic dictates that magnetic media can no more protect itself from misuse than a padlock can lock itself.

Software protection must reside outside the actual storage media. The technique can then be made as tamper proof as deemed necessary. If one is clever enough, patent law can be brought to bear on the method.

Software protection is at a crossroads and the choices are clear. You can give product away to a segment of the market, or take a stand against the theft of your intellectual property.

"...giving your software away is fine..."

We strongly believe that giving your software away is fine, if you make the decision to do so. However, if the public's sense of ethics is determining company policy, then you are no longer in control.

We have patented a device that protects your software while allowing unlimited archival copies and uninhibited use of hard disks and LANs. The name of this product is The BLOCK™.

The BLOCK is the only patented method we know of to protect your investment. It answers all the complaints of reasonable people concerning software protection.

In reality, the only people who could object are those who would like the option of stealing your company's product.

"...eliminating the rationale for copy-busting..."

Since The BLOCK allows a user to make unlimited archival copies the rationale for copy-busting programs is eliminated.

The BLOCK is fully protected by federal patent law rather than the less effective copyright statutes. The law clearly prohibits the production of work-alike devices to replace The BLOCK.

The BLOCK attaches to any communications port of virtually any microcomputer. It comes with a unique customer product number programmed into the circuit.

The BLOCK is transparent to any device attached to the port. Once it is in place users are essentially unaware of its presence. The BLOCK may be daisy-chained to provide security for more than one software package.

Each software developer devises their own procedure for accessing The BLOCK to confirm a legitimate user. If it is not present, then the program can take appropriate action.

"...possibilities... limited only by your imagination..."

The elegance of The BLOCK lies in its simplicity. Once you understand the principle of The BLOCK, hundreds of possibilities will manifest themselves, limited only by your imagination.

Your efforts, investments and intellectual property belong to you, and you have an obligation to protect them. Let us help you safeguard what's rightfully yours. Call today for our brochure, or a demo unit."

Software Security inc.
870 High Ridge Road Stamford, Connecticut 06905
203 329 8870

very simple to code, yet it can cause Paradox to perform vast amounts of work with very good performance.

The next step beyond the facilities provided by such products are those embodied in the program and application generators that are starting to appear. I have already mentioned Paradox; new products are here from Cosmos (Advanced Revelation) and coming from Barrington (the next version of Clarion). Many more are sure to appear because interest in these environments is not just academic. Professional programmers, especially those working for clients, know that the human labor component of software development is the most expensive. Anything that allows them to reduce costs by reducing time also allows them to lower prices while maintaining or raising profit margins. More importantly, reducing programming time leaves more time to think about the problem and, with luck, allows the developer to offer more creative or useful solutions to the business problem at hand.

By contrast, this is just not where C is positioned. Not saying that there isn't a place for C—I think there is—but C is problematic for the vast majority of business solutions that you and I want to deploy. It is complicated, time consuming, and unfriendly.

We have an interesting opportunity at hand to gauge the effectiveness of the new generation of development tools. OS/2 brings a new perspective to the fore because, as a new environment, it is quite naked. While hundreds of utilities and libraries exist for DOS that help the traditional programmer obtain function and reduce coding time, none yet exists for OS/2.

However, those same facilities exist *within* such tools as Clarion and most data managers. When Microrim demonstrated R:BASE running in OS/2 last April, all of its capabilities were present (although it did not fully exploit OS/2). What this means is that applications developers who have been using R:BASE can port their existing applications to OS/2 immediately and can begin work on new applications immediately, because all the resources of R:BASE will be present. Later, when R:BASE and other like products do exploit OS/2, the programmer gets the advantages of OS/2 without suffering the consequences of development time. The unfortunate souls building applications from scratch in C will have to wade through every OS/2 swamp that they once traversed in DOS.

WHAT'S IN THE BOX?

Since COMDEX last November, when IBM announced availability of OS/2 for December 4, we have all been waiting to see the real thing, which arrived quite promptly after the promised date.

The package itself sports some rather interesting differences from its DOS predecessors. The most obvious is the IBM Program License Agreement, which is no longer printed on the sleeve cover under the shrink-wrap but is contained in an open envelope attached to the *outside* of the shrink-wrap. I thought, perhaps, that IBM was trying a new twist on licensing designed to overcome the recent court decision in Louisiana (Vault vs. Quaid) in which so-called shrink-wrap licenses were, in effect, declared invalid. In the opinion of attorney and contributing editor Max Stul Oppenheimer, having the license on the outside of the package is no different from a legal standpoint than having it under the plastic. He suggested a mechanical reason: the agreement simply may have grown too large to fit conveniently on the sleeve cover. An IBM spokesman said that the company had no special reason for the change, except that the new agreement is easier to read and that IBM encourages its customers to read the license terms before opening the software product. The additional manufacturing expense represented by the externally attached envelope and separate sheet of paper for the license makes me believe that IBM does have some other reason for the change, however.

Oppenheimer also mentioned that IBM's licensing position might now be weakened slightly because having the license loose on the outside of the package raises the possibility that a package can be delivered without the purchaser having seen a copy of the license. Although the license is present inside the package in several places, the purchaser cannot possibly see it without breaking the seal.

The license language is somewhat different (Oppenheimer will report on the changes in a future issue). However, one very obvious addition to the terms falls in the "Limitations of Remedies" section, in which IBM limits its liability for actual damages to the "greater of $5,000 or the money paid

to IBM." The DOS 3.3 agreement limits remedies to a refund of the price paid, so IBM seems to be saying that it will take more responsibility for OS/2 than it did for DOS. Although IBM's language implies a ceiling of $5,000 per copy, the IBM spokesman said that it meant $5,000 or the total paid to IBM *per claim*. A larger company with, for example, 1,000 copies of OS/2 would be limited to whatever it had paid to obtain all its copies of the product, not $5 million. A curious side effect is that an individual purchasing one copy of OS/2 might obtain remedy in the amount of $5,000 (or 15 times the purchase price), while a larger buyer is, in effect, limited to the cost of the goods purchased. Oppenheimer said that such an interpretation is, in fact, inconsistent with the view that the license applies to a single machine, not to a site with multiple machines.

The OS/2 sleeve cover reveals some more interesting items. Whereas the covers for DOS list "Software Included" and "System Requirements," which are self-explanatory, OS/2 lists three new categories: warranty, program services, and service expiration date. The OS/2 warranty for both media and program is three months. Program services are available at no charge after the warranty expires, but the program services themselves expire on January 15, 1989, after which no service is available. This sounds like a full-service warranty on the programs until 1989; after that, IBM has no obligation to fix anything, so the user's only recourse is to buy a new version. The IBM spokesman explained that the difference between the warranty period and the service period was a legal one, that the warranty was a legal obligation, whereas the service was a decision on the part of IBM. The spokesman was emphatic in stating that IBM would provide the same level of excellent service throughout both the warranty and the service periods.

Inside the sleeve are two manuals, four 3.5-inch, 1.44MB diskettes (or the same number of 5.25-inch, 1.2MB diskettes), two cards, and two brochures. The very thin, spiral-bound *User's Guide* mostly discusses installation. The *User's Reference*, at 300-plus pages, is more or less equivalent to the manual that comes with PC-DOS and includes the reference section for OS/2 commands. The format of the manual is strikingly different from IBM's DOS manuals. One particularly strange twist to the documentation is the inclusion

THE PROGRAMMER'S SHOP

helps save time, money, and cut frustrations. Compare, evaluate, and find products.

Your Complete Tool Set: Only $50 per Month!

You could easily spend $1,500 pulling together the best programming environment. Now you can lease your choice of the best compiler, editor, debugger, and general purpose library, or whatever.

If your 2-year-old business generates $75K+ per year in revenue, call about our software-only lease.

Another innovation from The Programmer's Shop.

Call a Software "Leasing Specialist."

C Programmers: Complete Your Development Toolset.

Dbase translation, fussy preprocessing, graphics windowing, ISAMs: C tools cover the spectrum.

Select from the professional tools at right; or let us help you choose from over 100 others.

Call one of our "tech reps" TODAY.

Order before February 29, 1988 and mention "PT288" for these SPECIAL PRICES:

	List	Normal	SPECIAL
Ctree by Faircom	$395	$315	**$289**
dBx Translator	$350	$299	**$279**
Essential Graphics	$250	$185	**$165**
PC Lint	$139	$119	**$ 89**
Windows for Data	$295	$239	**$209**

386 Development Tools

386 Assembler/Linker	PC	$ 389
386 Debug - by Phar Lap	PC	$ 129
386/DOS Extender	PC	$ 919
DESQview PS/2	PC	$ 109
F77L-EM - by Lahey	MS	Call
High C - by Metaware	PC	Call
OS/286 & 386 by AI Architects	PC	Call

AI Languages

APT - Active Prolog Tutor - build applications interactively	PC	$ 49
ARITY Prolog - full, 4 Meg Interpreter - debug, C, ASM	PC	$ 229
COMPILER/Interpreter-EXE	PC	$ 569
Cogent Prolog Compiler	MS	$ 179
MicroProlog Prof. Comp./Interp.	MS	$ 439
PC Scheme LISP - by TI	PC	$ 85
Star Sapphire	MS	$ 429
TransLISP - learn fast	MS	$ 79
TransLISP PLUS	MS	$ 149
TURBO PROLOG by Borland	PC	$ 69

Others: IQ LISP ($239), IQC LISP ($269)

Basic

BAS_C - economy	MS	$ 179
BAS_PAS - economy	MS	$ 135
Basic Development Tools	PC	$ 89
db/Lib	MS	$ 119
Exim Toolkit - full	PC	$ 45
Finally - by Komputerwerks	PC	$ 85
Inside Track	PC	$ 49
Mach 2 by MicroHelp	PC	$ 55
NetWorks by Exim	PC	$ 89
QBase - screens	MS	$ 79
QuickBASIC	PC	$ **69**
Quick Pak-by Crescent Software	PC	$ 59
Quick-Tools by BC Associates	PC	$ 109
Stay-Res	PC	$ 59
True Basic	PC	$ 79
Turbo BASIC - by Borland	PC	$ 69
Turbo BASIC Database Toolbox	MS	$ 69

FEATURES

XQL - SQL for Btrieve callable from BASIC, C, and Pascal or for interactive query. Computed fields, specify sort order, manipulate composite records from joined files. No royalties. MS $459

Instant-C/16M - Addresses up to 16M for program and data. Incremental compilation makes development faster than Turbo C (compile and relink XLISP in 4 secs vs 24). 286/386 only. PC, List: $895

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. Formats: 3" laptop now available, plus 200 others. UPS surface shipping add $3/item.

RECENT DISCOVERY

SofTRAN, the Translation and Text Language by TransOptima - full procedural language like C plus pattern and nonprocedural constructs cuts development effort by up to 16 times. PC $349

C Language-Compilers

AZTEC C86 - Commercial	PC	$499
C86 PLUS - by CI	MS	$359
Datalight Optimum - C	MS	$ 99
Lattice C - from Lattice	MS	$259
Microsoft C 5.0- Codeview	MS	**$275**
Microsoft Quick C	MS	$ 67
Rex - C/86 standalone ROM	MS	$695
Turbo C by Borland	PC	$ 67

C Libraries-Files

BTree by Soft Focus	MS	$ 69
CBTREE - Source, no royalties	MS	$ 99
ctree by Faircom - no royalties	MS	$315
rtree - report generation	PC	$239
dB2C Toolkit V2.0	MS	$249
dbQUERY - ad hoc, SQL-based	MS	Call
dbVISTA - Object only	MS	Call
Source - Single user	MS	Call
dBx - translator	MS	$299

C-Screens, Windows, Graphics

C Worthy Interface Library	PC	$249
Curses by Aspen Scientific	PC	$109
dBASE Graphics for C	PC	$ 69
ESSENTIAL GRAPHICS - fast	PC	$185
FontWINDOW/PLUS	PC	$229
GraphiC - new color version	PC	$279
Greenleaf Data Windows	PC	$155
w/source	PC	$259
Terminal Mapping System	PC	$279
TurboWINDOW/C - for Turbo C	PC	$ 75
View Manager - by Blaise	PC	$199
Windows for C - fast	PC	$149
Windows for Data - validation	PC	$219
Vitamin C - screen I/O	PC	$159
VC Screen	PC	$ 79
ZView - screen generator	MS	$149

Atari ST & Amiga

We carry full lines of Manx & Lattice.

DBASE Language

Clipper compiler	PC	$399
dBASE II	MS	$329
dBase III Plus	PC	$429

Call for a catalog, literature, and *solid* value

800-421-8006

THE PROGRAMMER'S SHOP™

Your complete source for software, services and answers

5-P Pond Park Road, Hingham, MA 02043
Mass: 800-442-8070 or 617-740-2510

RECENT DISCOVERY

FORCE III, Dbase Compiler by Sophco - small .EXEs, user-defined functions, I/O directives through BIOS/DOS/ANSI/FORCE/user-defined, extensions include FOR..NEXT loops, soundex, 1D arrays. Maverick. PC $109

DBASE Language Cont.

dBASE III LANPack	PC	$649
DBXL Interpreter by Word Tech	PC	$ 99
FoxBASE+ Dev. - V2.0	MS	$289
Quicksilver by Word Tech	PC	$369

DBASE Support

dAnalyst	PC	$ 89
dBase Tools for C	PC	$ 65
dBrief with Brief	PC	Call
dBC III by Lattice	MS	$169
Documentor - dFlow superset	MS	$229
Genifer by Bytel-code generator	MS	$279
QuickCode III Plus	MS	$239
R&R Report Writer	MS	$139
Seek-It - Query-by-example	PC	$ 79
Silver Comm Library	MS	$139
Tom Rettig's Library	PC	$ 79
UI Programmer - user interfaces	PC	$249

DataBase & File Management

CQL	PC	$ 359
DataFlex by Data Access	PC	$ 899
DataFlex multiuser	PC	$1149
Magic PC	PC	$ 699
Paradox - original	PC	$ 369
Paradox V2.0	PC	$ 469
Revelation by Cosmos	PC	$ 779

Multilanguage Support

BTRIEVE ISAM	MS	$185
BTRIEVE/N-multiuser	MS	$455
GSS Graphics Dev't Toolkit	PC	$375
HALO Development Package	MS	$389
Graphics	PS	$209
Help/Control - on line help	PC	$ 99
Hoops Graphics Library	PC	$549
Instant Programmer's Help	MS	$ 79
Informix 4GL-application builder	PC	$839
Informix SQL - ANSI standard	PC	$699
NET-TOOLS - NET-BIOS	PC	$129
Opt Tech Sort - sort, merge	MS	$ 99
Norton Guides	PC	$ 75
Panel Plus	MS	$395
Pfinish - by Phoenix	MS	$229
Report Option - for Xtrieve	MS	$109
Screen Sculptor	PC	$ 89
SSP/PC - 145+ math routines	PC	$269
Synergy - create user interfaces	MS	$329
Xtrieve - organize database	MS	$199
ZAP Communications - VT 100	PC	$ 89

CIRCLE NO. 162 ON READER SERVICE CARD

of what IBM calls "command diagrams," which are flow schematics that illustrate the syntax and use of OS/2 commands. They will be easy to read for those of us with a programming background, but many end users may find them scary. The diagrams do, however, more precisely define the allowable syntax for each command.

The two cards are a proof of license and an order card for a free update to version 1.1 of OS/2—the one with the Presentation Manager. The brochures are a four-page "Read This First" overview document and a seven-page document that is called "License Information," which details the terms of the warranty and lists the specifications, presumably as grounds for determining which of the service requests will be honored.

Several programs are provided that are, in the manual's words, " . . . intended to be used with assistance from an IBM Service Representative." This wording strongly implies that the OS/2 user has someone to call upon for assistance with system difficulties. The "License Information" brochure describes how to obtain service; the IBM spokesman confirmed my interpretation of IBM's language, which is to call the place where you purchased your copy. Larger sites that purchase directly from IBM can obtain service from IBM's support center. One of the programs, CREATEDD, puts a complete system dump onto diskettes, referred to as "customer service diskettes," that can be used by an IBM representative for problem determinations—shades of IBM OS/360 ABEND dumps. Welcome to the desktop mainframe.

A quick look at the supplied utilities and internal commands reveals a list that almost exactly matches the ones delivered with DOS. Several commands and programs have been added to support certain new capabilities of OS/2, but nothing more exotic than CHKDSK or DISKCOPY.

PC Tech Journal also received evaluation copies of IBM's C/2 and MASM/2 products. That led us to wonder whether we had enough software and documentation to develop an OS/2 program. At the very least, a compiler, linker, and OS/2 libraries are needed.

As with DOS, the linker, LINK.EXE, is included with OS/2 as well as with both language products. Libraries come in two flavors in OS/2: .LIB (regular and implicit libraries) and .DLL (dynamic link libraries). A long list of .DLLs on OS/2 contain not only application interface code but also routines that OS/2 uses, so those must always be included with OS/2, not a language or a toolkit. However, languages need to be linked with a .LIB-style library so that the .EXE program file will know what .DLLs to dynamically link at runtime. That library, DOSCALLS.LIB, also comes with OS/2.

What this means is that the basic interface library for OS/2 is included with every copy of OS/2, and that means a third-party language supplier can rely on the presence of a common tool—one more feature that does not have to come with the language. IBM should continue this practice by supplying the basic libraries and resources for the Presentation Manager along with OS/2 1.1, but the fact is that a toolkit for that version will be a necessity for Presentation Manager developers. According to IBM, the OS/2 1.1 Toolkit will contain a font editor, a dialog editor, a bit-map/cursor editor, a resource compiler, and sample Presen-

tation Manager programs. This is the same collection of items that Microsoft delivers with its Windows Software Development Kit.

Technical information is another story. The OS/2 manual has no developer information, which is consistent with the way IBM supplied the more recent versions of PC-DOS. The *OS/2 Technical Reference*, which is a separate purchase of $200, clearly will be necessary for anyone doing serious programming in OS/2. However, IBM has introduced a "gotcha": the *Technical Reference* is a *prerequisite* purchase for the IBM OS/2 Programmer Toolkit, which is necessary for developers wishing to create family API programs and those wishing to construct their own dynamic link libraries; it costs $750.

So if you want to develop a C program in OS/2 and you somehow know all the system calls, you will need a copy of the operating system ($325) and a copy of C/2 ($525) for a total of $850. If you need to know about system calls, you need the *OS/2 Technical Reference*, which brings the total to $1,050. The family API or a desire to build .DLLs necessitates the toolkit, and the total rises to $1,800. Finally, the MASM/2 assembler for those little time-critical routines brings the total to a nice, round $2,000. That pricing at least explains why Microsoft charged $3,000 for the MS-OS/2 toolkit, which included not only all of the items above but a subscription to DIAL, a three-day seminar, the Presentation Manager specification, and a copy of Windows and the Windows Software Development Kit.

Even with the inclusion of the basic libraries, the OS/2 box is still fairly empty. The high price of OS/2 buys only multitasking and access to more memory; until there is an application to use those facilities, OS/2 is a big investment. Perhaps version 1.1, with its Presentation Manager, will fill the box a bit better.

On a more positive note, at least the product is here and backed by IBM. That gives developers a stable base to begin OS/2 research and possible development.

As a final aside, technical editor Ted Mirecki ran OS/2 on his AT 339 and on a PS/2 Model 60. I tried it on my Compaq 386/20. No problems surfaced. We were delighted to see the PS/2 running because the OS/2 included in Microsoft's toolkit never did; this is the first time OS/2 has run on a PS/2 in our offices.

Discover Parallel Processing!

Monoputer™

The Most Cost Effective Transputer Development System

MicroWay's **Monoputer** is the best selling Transputer-based PC coprocessor in the world. It was the first board available to run the 20 MHz T414 or T800. As a result, it received many rave reviews in the UK (available on request) and became the standard Transputer software development tool. Parallel code can be executed on a single Monoputer or on an array of Monoputers wired together by their external link lines. The Monoputer includes 2 megabytes of 100 nsec DRAM, a 20 MHz T414 or T800 and the MicroWay stand alone Occam Compiler, which generates Transputer code that runs under MS-DOS. Optional tools include our licensed version of the TDS and a Pascal, Fortran, C, and Prolog.

Quadputer™

Mainframe Power For Your PC!

MicroWay's **Quadputer** is the most versatile multiple Transputer coprocessor on the market today. It can be purchased with 2, 3, or 4 Transputers, each of which has 1 or 4 megabytes of local memory. Two or more Quadputers can be easily cabled together to build larger parallel processor systems. A single Quadputer using four T800s provides 40 MIPS of CPU and 6 megaflops of NDP throughput at one fiftieth the cost of a comparably performing mainframe.

Attend MicroWay's Seminar on Parallel Processing, April 6th in Plymouth, Massachusetts! For more information please call (617) 746-7341. After July, 1988, call (508) 746-7341.

MicroWay

The World Leader in PC Numerics

P.O. Box 79, Kingston, Mass. 02364 USA (617) 746-7341
32 High St., Kingston-Upon-Thames, UK 01-541-5466
St. Leonards, NSW, Australia 02-439-8400

TECH RELEASES

The latest in hardware, software, and technology for systems developers and integrators.

Workstation² Series from Unisys

SYSTEMS

Tandon Computer Corporation has announced the **Tandon 386**, a microcomputer featuring a 20-MHz 80386 with 2MB RAM on the system board; a 112MB internal Winchester disk drive with a 27-ms access time; and a receptacle for the Tandon Personal Data Pac, which creates a transportable computing environment. The internal Winchester disk drive uses Tandon's high-performance run length limited (RLL) controller that features a built-in 128KB disk cache for reducing disk access time. The Personal Data Pac is a lightweight, compact 30MB packaged drive that can also run in Tandon's PAC 286 and its Ad-Pac subsystem, which attaches by shielded cable and controller card to any PC. The Tandon 386 also includes a socket for an 80387, a realtime clock, and a 1.2MB 5.25-inch diskette drive. It has a 64KB static RAM cache to allow faster memory access. Tandon 386, $9,999; 30MB Personal Data Pac, $399; Ad-Pac, $599.
Tandon Computer Corporation, 405 Science Drive, Moorpark, CA 93021; 805/523-0340
CIRCLE 303 ON READER SERVICE CARD

Tandon 386 from Tandon Computer Corp.

An 80386-based 20-MHz portable computer has been introduced by **Datavue Corporation.** The machine, known as the **Smoke386**, has standard ports that include composite, external VGA monitor, external 5.25-inch diskette drive, a parallel port, and two serial ports. Smoke386's memory ranges from 256KB to 1.25MB. Several disk drive configurations are available: dual 3.5-inch 1.44MB diskette drives; one 40MB internal hard-disk drive with one 3.5-inch 1.44MB diskette drive; and one 5.25-inch diskette drive. Smoke386 is AC-only powered and uses a black on white (or inverse) VGA-type (640-by-480 pixel resolution) display. Smoke386's screen outline is square and has an aspect ratio of 1:1. An expansion box will be made available for Smoke386 to accommodate two full 8-bit or 16-bit expansion cards. Price is not yet available.
Datavue Corporation, One Mecca Way, Norcross, GA 30093-2919; 404/564-5555
CIRCLE 302 ON READER SERVICE CARD

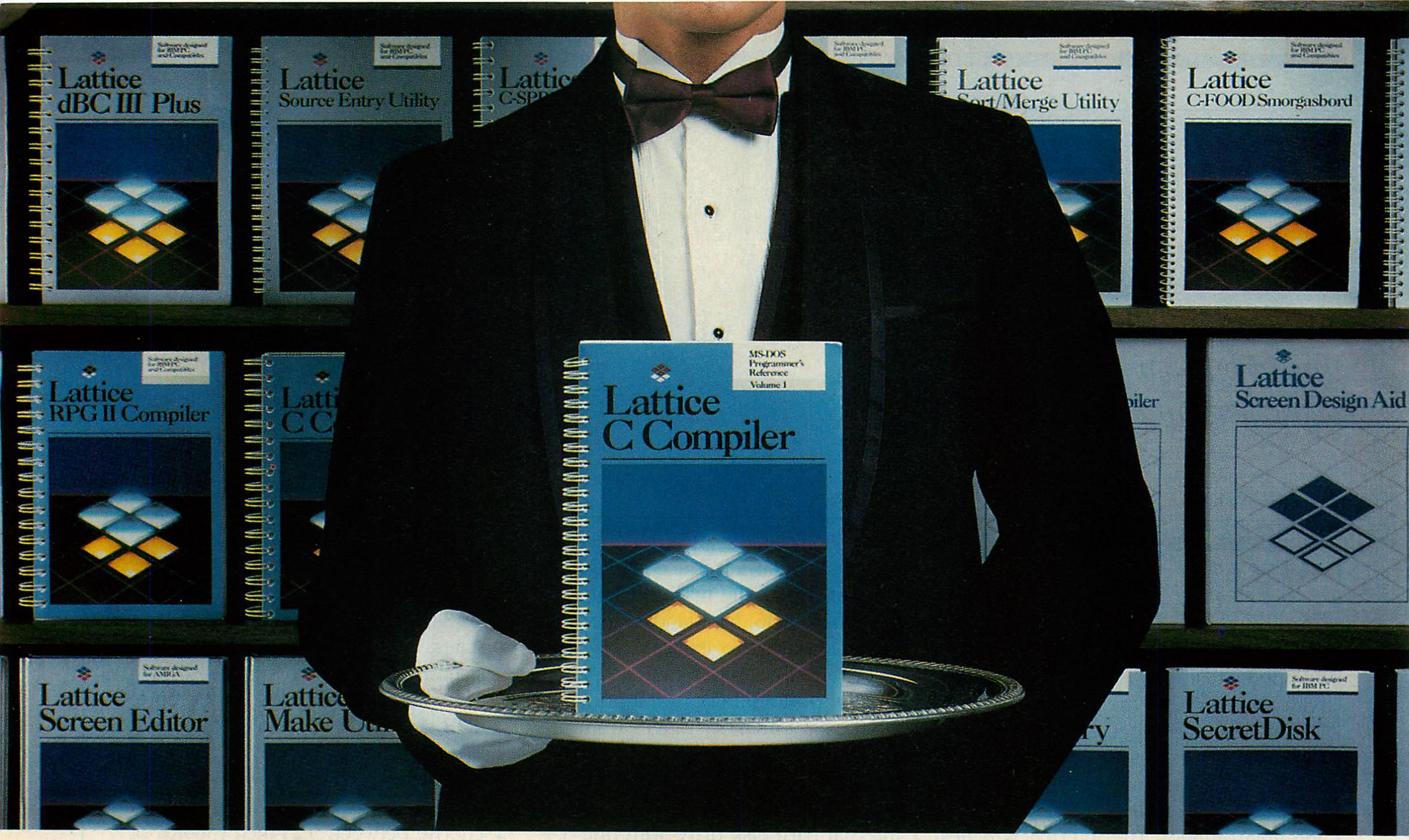
A family of personal computers compatible with MS-DOS, XENIX, Windows, and Microsoft OS/2 operating systems has been introduced by **Unisys Corporation**. These microcomputers can be configured to user needs. The **Personal Workstation² Series 800/16/20** comes standard with either a 16- or 20-MHz, zero-wait-state Intel 80386; 1MB internal memory; serial port and parallel ports; seven full-length slots (two 8-bit, four 16-bit, and one 32-bit); a socket for an 80387; and MS-DOS 3.3. Options include a 5.25-inch (360KB or 1.2MB) or 3.5-inch (720KB or 1.44MB) diskette drive. Mass storage options include 40MB, 69MB, or 116MB hard-disk drives, all of which have a 30-ms average access time and a 60MB tape drive. Prices range from $4,780 for the base configuration to $8,085 for machines with the 116MB hard disk.

The **Personal Workstation² Series 500/12** includes a choice of a 6-MHz Intel 80286 with one wait state, an 8-MHz 286 with zero or one wait states, or a 12-MHz 286 with one wait state. Internal memory of 640KB is standard and can be expanded up to 4MB. Built-in features include a parallel port, asynchronous/synchronous serial communications port, two 8-bit slots, and three 8/16-bit slots. The user can choose between 5.25-inch (360KB or 1.2 MB), or 3.5-inch (720KB or 1.44MB) diskette drives. Mass storage options are 20MB or 40MB hard-disk drives with 40- or 30-ms access times, respectively. Prices range from $2,495 for the base configuration to $4,585 for a machine with the 40MB hard disk.

The **Personal Workstation² Series 300/10** includes either an 8- or 10-MHz, one-wait-state 286 with 640KB internal memory (expandable to 1.5MB), serial port and parallel port, monochrome/EGA controller, and two 8/16-bit slots. The user can select between 5.25-inch (360KB or 1.2MB) or 3.5-inch (720KB or 1.44 MB) diskette drives. A 20MB hard-disk drive with a 40-ms access time is available. A total of two half-height peripherals can be added. Prices range from $1,305 for base configuration to $2,380 with the 40MB hard disk.
Unisys Corporation, PO Box 500, Blue Bell, PA 19424-0001; 800-547-8362
CIRCLE 301 ON READER SERVICE CARD

CONNECTIONS

A Micro Channel converter board that serves as a host adapter from the Micro Channel to the PC bus for the IBM PC has been announced by **Advanced Digital Corporation** (ADC). The **TransFormer 2** enables IBM PC owners to use most existing PC board interface cards with the PS/2 Micro Channel for complete system compatibility.


Lattice
dBC III Plus
Lattice
Source Entry Utility
Lattice
Sort/Merge Utility
Lattice
C-FOOD Smorgasbord
Lattice
RPG II Compiler
Lattice
Screen Design Aid
MS-DOS
Programmer's
Reference
Volume 1
Lattice
C Compiler
Lattice
Screen Editor
Lattice
SecretDisk

Genesys Solution LAN package from ALR

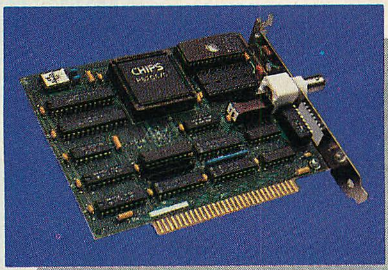
The Brooklyn Bridge from White Crane Systems

The TransFormer 2 fits into the PS/2 Model 50, 60, or 80 and is connected by a cable to the PC expansion bus. The product includes the TransFormer 2 board, cable, and the ADC XI card (a PC expansion interface board). The TransFormer 2 can also be used with ADC's Personal Network software, Novell's NetWare, and, in the future, with OS/2. It offers multipurpose compatibility with most other board enhancements such as RS-232 and Ethernet and it can also run direct memory access (DMA). $695.
Advanced Digital Corporation, 5432 Production Drive, Huntington Beach, CA 92649; 714/891-4004
CIRCLE 313 ON READER SERVICE CARD

The **Genesys Solution**, a complete 4-to-60 user LAN package with 80286 diskless workstations, has been announced by **Advanced Logic Research, Inc.** (ALR). The network system is based on ALR's 386/220 system running at 20 MHz and includes 2MB of 32-bit memory standard. The system has a 42MB, 28-ms hard disk and the controller, which buffers a full track of data, and it performs with a 1:1 interleave factor. The ALR **Micronode** diskless workstation is an 8-MHz, 286-based desktop workstation that has a compact size (7.75 inches high by 5.25 inches wide by 16 inches deep) and comes standard with 512KB of RAM (expandable to 1MB), an ALR ARCnet-compatible adapter card, a high-resolution monographics adapter, and tilt/swivel monochrome monitor. An **ALR Micronode disk-pack**, which includes a 1.2MB 5.25-inch diskette drive and a 40MB, 28-ms Winchester disk drive and controller, is an available option. Genesys Solution (four-user), $9,995; Micronode, $1,195; disk-pack, $1,495.
Advanced Logic Research, Inc., 10 Chrysler, Irvine, CA 92718; 714/581-6770
CIRCLE 311 ON READER SERVICE CARD

A very compact 3270 protocol converter has been announced by **Local Data Inc.** Small enough to mount inside the majority of printers, plotters, and other hardcopy devices, the **InterLynx/4110** is based on very large scale integration (VLSI) technology. Total throughput of the InterLynx/4110 exceeds 2,200 characters per second (cps) or 40 pages per minute (ppm) in nonimpact printers, and 1,500 lines per minute (lpm) in line printers. The InterLynx/4100 is based on the DP8344 BiPhase Communications Processor (BSP) which was recently introduced by National Semiconductor. In OEM quantities, $300 each.
Local Data, Inc., 2771 Plaza Del Amo, Torrance, CA 90503; 213/320-7126
CIRCLE 317 ON READER SERVICE CARD

A 3270 emulation board that is based on Chips and Technologies' CHIPSlink, a single-chip protocol controller for 3270 terminal emulation, has been announced by **Quadram Corporation**. Using only 16 integrated circuits, **Main-**



MainLink II, an emulation board from Quadram

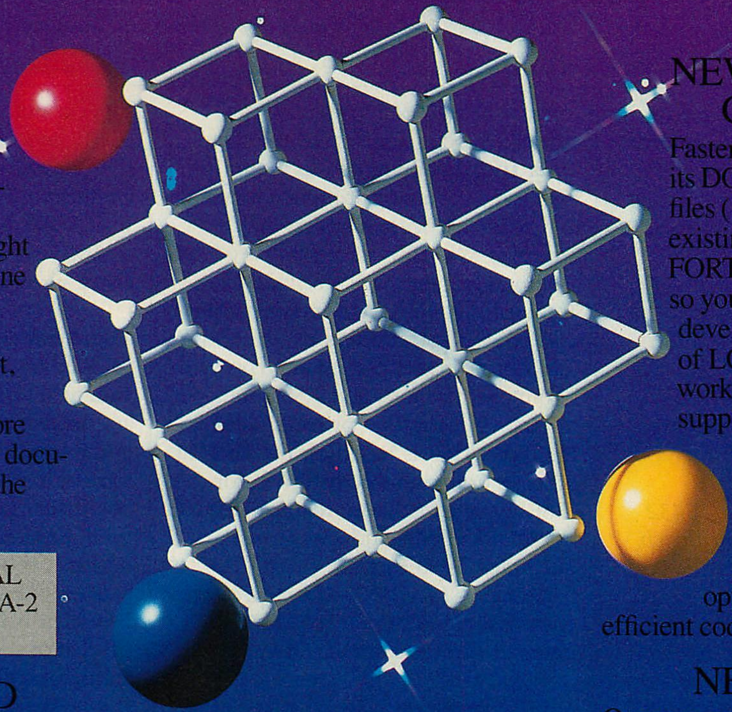
Link II features include low power consumption as well as enhanced reliability. The board will be fully compatible with IBM PS/2 computers due to CHIPSlink's ability to emulate 3270 terminals by pairing the controller chip with its companion Micro Channel interface component. In addition, MainLink II will be compatible with IBM and IRMA program interfaces and will support both control unit terminal (CUT) and distributed function terminal (DFT) modes. $399.
Quadram Corporation, One Quad Way, Norcross, GA; 30093-2919; 404/923-6666
CIRCLE 310 ON READER SERVICE CARD

The Brooklyn Bridge, originally developed by **White Crane Systems Inc.** to transfer files between desktop and laptop computers, has been upgraded to transfer files between desktop computers and IBM PS/2 microcomputers as well. With a universal cable that features 9- and 25-pin connectors on both ends, the dual connectors enable a single package of The Brooklyn Bridge to be used on numerous laptops, desktops, and the PS/2 family of computers. The Brooklyn Bridge consists of RAM-resident, file-transfer software on a 5.25-inch diskette drive for desktop computers, a 3.5-inch disk for laptops or PS/2 microcomputers, and a null-modem serial cable to connect the two computers. It can also be used to transfer files from a microcomputer to a printer, tape back-up system, Bernoulli box, optical disk, and other peripheral devices. $129.95.
White Crane Systems, Inc., Suite 151, 6889 Peachtree Industrial Blvd., Norcross, GA 30092; 404/394-3119
CIRCLE 309 ON READER SERVICE CARD

Version 2.0 of **TOPS Local Area Network** has been introduced by **TOPS,** a Sun Microsystems Company (formerly Centram). The TOPS LAN is a distributed file server that provides transparent file sharing among Apple Macintosh computers, the IBM PC and compatibles, and UNIX-based systems. **TOPS/DOS version 2.0** will allow PCs on the TOPS Network to access dedicated printers formerly available to only one user. TOPS/DOS also includes Flash/Talk, a PC-to-PC communications architecture that operates on TOPS at

SOFTWARE ENGINEERING COMES OF AGE.

ANNOUNCING LOGITECH MODULA-2 VERSION 3.0

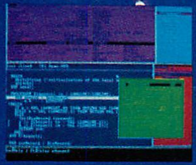


Modula-2 is the language of choice for modern software engineering, and LOGITECH Modula-2 is the most powerful implementation available for the PC. The right language and the right tools have come together in one superior product. Whether you're working on a small program or a complex project, with LOGITECH Modula-2 Version 3.0 you can write more reliable, maintainable, better documented code in a fraction of the time at a fraction of the cost.

FREE TURBO PASCAL TO LOGITECH MODULA-2 TRANSLATOR

NEW, IMPROVED DEBUGGERS

Time gained with a fast compiler can be lost at debug time without the right debugging tools. With the powerful Logitech Modula-2 Debuggers you can debug your code *fast*, and dramatically improve your overall project throughput.



The Post Mortem Debugger analyzes the status of a program after it has terminated while the dynamic, Run Time Debugger monitors the execution of a program with user-defined breakpoints. With their new, mouse based, multiple-window user interface these powerful debugging tools are a pleasure to use.

NEW, INTELLIGENT LINKER

Links only those routines from a particular module that you need, so you eliminate unreferenced routines and produce smaller, more compact executable files.

☐ LOGITECH Modula-2 V. 3.0 Compiler Pack **$99**
Compiler in overlay and fully linked form, Linkable Library, Post Mortem Debugger, Point Editor

☐ LOGITECH Modula-2 V. 3.0 Toolkit **$169**
Library sources, Linker, Run Time Debugger, MAKE, Decoder, Version, XRef, Formatter

☐ LOGITECH Modula-2 V. 3.0 Development System **$249**
Compiler Pack plus Toolkit

☐ Turbo Pascal to Modula-2 Translator **FREE**
With Compiler Pack or Development System

☐ Window Package **$49**
Build true windowing into your Modula-2 code.

☐ Upgrade Package
Call LOGITECH for information or to receive an order form.

Add $6.50 for shipping and handling. California residents add applicable sales tax. Prices valid in U.S. only.

Total Enclosed $________

☐ VISA ☐ MasterCard ☐ Check Enclosed

Card Number ________ Expiration Date ________

Signature ________

Name ________

Address ________

City ________ State ________

Zip ________ Phone ________

NEW, IMPROVED COMPILER

Faster and more flexible. Now its DOS linker compatible object files (.OBJ) can be linked with existing libraries in C, PASCAL, FORTRAN and ASSEMBLER—so you can build on previous development and put the power of LOGITECH Modula-2 to work for you right now. Fully supports Wirth's latest language definition, including LONGINT and LONGSET, which provides large set support including SET of CHAR. Provides optimization for tighter, more efficient code generation.

NEW EDITOR

Our new, mouse based editor is fully integrated, easy to learn, fast and easy to use, and very customizable. Its multiple, overlapping windows and color support make it easy to manage parts of one file or several files on the screen at one time. You'll love using it—with or without a mouse.

Call for information about our VAX/VMS version, Site License, University Discounts, Dealer & Distributor pricing.

To place an order call toll-free:
800-231-7717
In California:
800-552-8885

LOGITECH

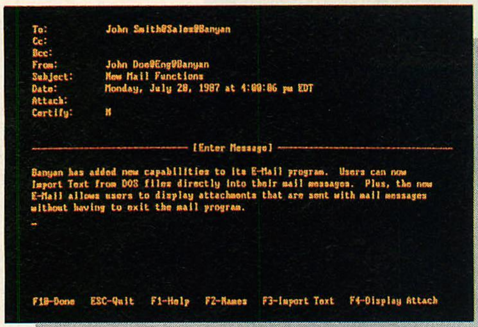
LOGITECH, Inc.
6505 Kaiser Drive, Fremont, CA 94555
Tel: 415-795-8500

In Europe: LOGITECH, Switzerland
Tel: 41-21-87-9656 Telex 458 217 Tech Ch

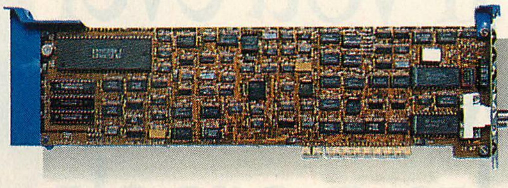
In the United Kingdom: LOGITECH, U.K.
Tel: 44908-368071 Fax: 44908-71751

Turbo Pascal is a registered trademark of Borland International. VAX and VMS are registered trademarks of Digital Equipment Corp.

CIRCLE NO. 229 ON READER SERVICE CARD



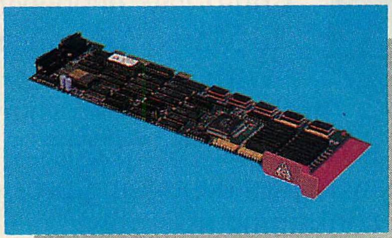
Screen from Banyan Vines, release 3.0



IDEAssociates' IDEAcomm 3278/DFT

three times its former AppleTalk speed. **TOPS/Macintosh version 2.0** introduces a "Remember" function that allows users to automatically make files available to the network and access remote files; it is fully compatible with AFP (Apple File Protocol) applications. Both TOPS/DOS and TOPS/Macintosh support AppleTalk zones and offer improved password protection. The new version of TOPS will recognize version 2.0 only. TOPS/DOS, $189; TOPS/Macintosh, $189; upgrades per node (until March 15, 1988), $29.
TOPS, 2560 Ninth Street, Suite 220, Berkeley, CA 94710; 415/549-5900
CIRCLE 314 ON READER SERVICE CARD

An addition has been made to **Orchid Technology**'s family of networking products. Developed for use in the IBM Micro Channel architecture, **PCnet-2** provides owners of PS/2 Models 50 and 60 with a LAN card that is compatible



Orchid Technology's PCnet-2

with other PCnet and PCnet/Conquest cards in the same network. Some features specific to the Micro Channel bus of the PS/2 include position option select (POS), burst-mode direct memory access (DMA), and shared interrupts and DMA. Available software includes PCnet software that is now compatible with DOS 3.3, NETBIOS, and PCnet/Advanced Netware. Per card, $495.
Orchid Technology, 45365 Northport Loop West, Fremont, CA 94538; 415/683-0300
CIRCLE 315 ON READER SERVICE CARD

Release 3.0 of VINES, a virtual networking operating system, has been announced by **Banyan Systems Inc.** This version will offer users significant performance improvements, mainframe-like security, better reliability, more system administration and management features, and full DOS 3.3 support. Two Transport Control Protocol/Internet Protocol (TCP/IP) options are available: one provides server-to-server communications and the other allows a Banyan network server to act as an IP router. Support for two new LANs, Western Digital's StarCard Plus and Micom Interlan's NI5210, has been included. Other features include improved mainframe and minicomputer connections, addition of asynchronous terminal emulation scripting, and enhancements to Banyan Mail, an electronic mail package. VINES's new scripting language is compatible with DCA/Crosstalk Communication's Crosstalk XVI scripts. Release 3.0 will allow printers attached to PCs on a VINES network to be shared by all network users, no longer requiring printers to be attached to dedicated print servers. Upgrades available at no charge to purchasers of VINES after September 15, 1987, and members of Banyan Systems Support program.
Banyan Systems Inc., 115 Flanders Road, Westboro, MA 01581; 617/898-1000
CIRCLE 312 ON READER SERVICE CARD

IDEAssociates, Inc. has announced two PC-to-mainframe communications products compatible with the IBM PC and compatibles and PS/2 Micro Channel architecture; both products offer multiple host sessions and provide enhanced file transfer capability. **IDEAcomm 3278/DFT** is a local coaxial PC-to-mainframe communications product that allows up to five host sessions, two of which many be used for display and printer emulation. **IDEAcomm 3270/SNA** provides remote communications via a synchronous modem for a maximum of eight host sessions that may be used for display and printer emulation. It provides emulation for the IBM 3174, 3274, 3276 remote controllers. The **IDEAFT**, a file-transfer program, allows both products to quickly and easily download and upload files between PC and mainframe under TSO, VM/CMS, and CICS host environments. The products also provide terminal emulation for the IBM 3278 Models 2, 3, 4, 5, and 3179, and allow both serial and parallel PC printers to emulate the IBM 3287 printer. The products feature 132-column support for the IBM 3278 Model 5, a hot-key capability allowing users to toggle between the PC and host operating systems without logging off the mainframe, a record and playback capability to store frequently used keystrokes, international-keyboard support, light-pen support, and a screen-capture feature. IDEAcomm 3278/DFT, $1,295; upgrades from 3278, $695; 3270/SNA, $895.
IDEAssociates, Inc., 29 Dunham Road, Billerica, MA 01821; 800/257-5027; 617/663-6878
CIRCLE 316 ON READER SERVICE CARD

PERIPHERALS

The **ARTIST 10/16 VGA**, a graphic controller for the IBM PS/2 computers, has been developed by **Control Systems**. Offering higher resolution than the VGA graphics shipped with the PS/2 systems, the ARTIST 10/16 VGA displays 16 colors at 1024-by-768 noninterlaced or 1024-by-1024 interlaced pixel resolution. It has 1MB of graphics memory and uses the Hitachi ACRTC graphics microprocessor. $2,995.
Control Systems, Inc., 2675 Patton Road, St. Paul, MN 55113; 800/826-4281; 612/631-7800
CIRCLE 308 ON READER SERVICE CARD

If you ever wanted to take a crack at assembly language, now's the time.

You probably already know that assembly language subroutines are the smartest way to get the fastest programs.

But if the complexities of working in assembler made you think twice, here's some good news. We've made Microsoft® Macro Assembler Version 5.0 a lot easier to use.

We eased the learning process by giving you the best support around. We completely revised our documentation. The new Mixed Language Programming Guide gives you step by step instructions for linking your assembly code with Microsoft QuickBASIC, C, FORTRAN, Pascal and other languages. And you get a comprehensive reference manual with listings of the instruction set and examples of each instruction. We didn't stop there, though. You also get an on-disk collection of templates and examples.

We've also dramatically simplified the high-level language interface. In just a few simple steps, you can be calling Macro Assembler subroutines from programs written in your favorite language.

Now that you're writing the fastest programs, Microsoft is giving you the fastest way to debug them. For the first time, we've added our CodeView® debugger to Macro Assembler.

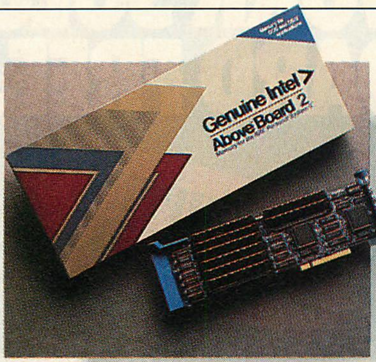
With source code and comments on your screen, Microsoft CodeView makes debugging programs containing assembly language subroutines a snap.

And you'll be glad to know that you don't sacrifice any speed for all the ease of use. We took the fastest Macro Assembler on the market and made it even faster.

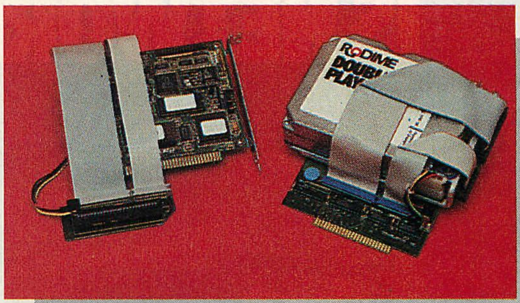
So what are you waiting for? Get your hands on Microsoft Macro Assembler and see what it's like to break your personal speed limit.

Microsoft®

For more information or for the name of your nearest Microsoft dealer, call (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada, call (416) 673-7638.



Intel's Above Board 2 memory board

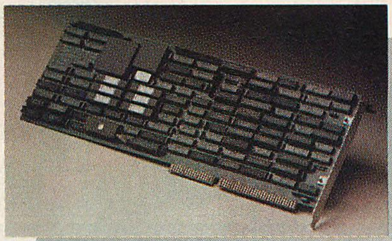


Rodime's Double Play hard-disk subsystem

A memory board that provides up to 2MB expanded and OS/2 memory in a single slot for the IBM PS/2 Models 50 and 60 has been announced by **Intel Corporation's Personal Computer Enhancement Operation (PCEO)**. The **Above Board 2** is compatible with the Lotus/Intel/Microsoft expanded memory specification (LIM EMS) version 4.0, as well as OS/2. For switching between EMS 4.0 and OS/2 memory, Above Board 2 offers an easy-to-use software program. Software utilities included are a RAM disk support program that allows users to store and quickly access very large programs or files, a diagnostic routine program, and a print buffer that allows the computer and printer to be in use at the same time. The board is based on single in-line memory modules (SIMMs.) 0KB version, $445; 512KB version, $645.
Intel Corporation, Mail Stop C03-07, 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-6497; 800/538-3373; 503/629-7354
CIRCLE 305 ON READER SERVICE CARD

A 45MB, 28-ms hard-disk subsystem for the IBM PS/2 Model 50 has been announced by **Rodime, Inc.** The **Rodime Double Play**, intended to replace IBM's original equipment 20MB hard disk, has a proprietary interface board included with each kit to permit dealers to salvage the removed OEM drive by installing it in the IBM Model 25. The Rodime Double Play features a 3.5-inch drive with an average access time of 28 ms. All cables, interface boards, controllers, and brackets are included to allow both its installation into the Model 50, and the installation of the removed 20MB drive into a Model 25. $899.
Rodime, Inc. Peripheral Systems Division, 29525 Chagrin Blvd., Suite 214, Pepper Pike, OH 44122; 216/765-8414
CIRCLE 306 ON READER SERVICE CARD

For upgrading 80286-based IBM computers to the 80386, **Aox Incorporated** has introduced a 20-MHz, 386 processor card. The **MASTER 386-20** features simple one-step installation, requiring no chips to remove, cables to connect, or software to install. The MASTER can accept an optional 80387, and will run in protected mode (as required by UNIX, The Software Link's PC-MOS/386, and Microsoft Windows/386). It can be equipped with true 32-bit memory using Aox's optional memory cards: a 2MB card expandable to 10MB, or a 4MB card expandable to 16MB. MASTER 386-20, $2,195; 2MB card, $1,250; 4MB card, $1,995.



Aox's Master 386 processor card

Aox Incorporated, 486 Totten Pond Road, Waltham, MA 02154; 617/890-4402
CIRCLE 304 ON READER SERVICE CARD

SOFTWARE DEVELOPMENT

Several products announced by **IBM Corporation** with the introduction of the Personal System/2 on April 2, 1987, are now available or the shipping dates have now been set. Already being shipped is its **Operating System/2 Standard Edition 1.0**, which offers extended memory addressability up to 16MB and allows users to run multiple concurrent application programs. The **IBM OS/2 Extended Edition**, to be available in July 1988, will offer the features of the Standard Edition plus an advanced relational database manager, a communications manager that provides intersystems communications, and improved connectivity and terminal emulation. OS/2 Standard Edition 1.0, $325; (upgrades from DOS 3.x, $200); OS/2 Extended Edition, $795; upgrade from Standard Edition 1.0, $545.

IBM Advanced Interactive Executive for Personal System/2 (AIX PS/2) is scheduled to ship in September 1988. AIX PS/2 is a 32-bit UNIX-based operating system for the IBM PS/2 Model 80 with a minimum of 2MB of memory. It provides multiuser, multitasking, and virtual memory support for up to 16 concurrent users. $595.

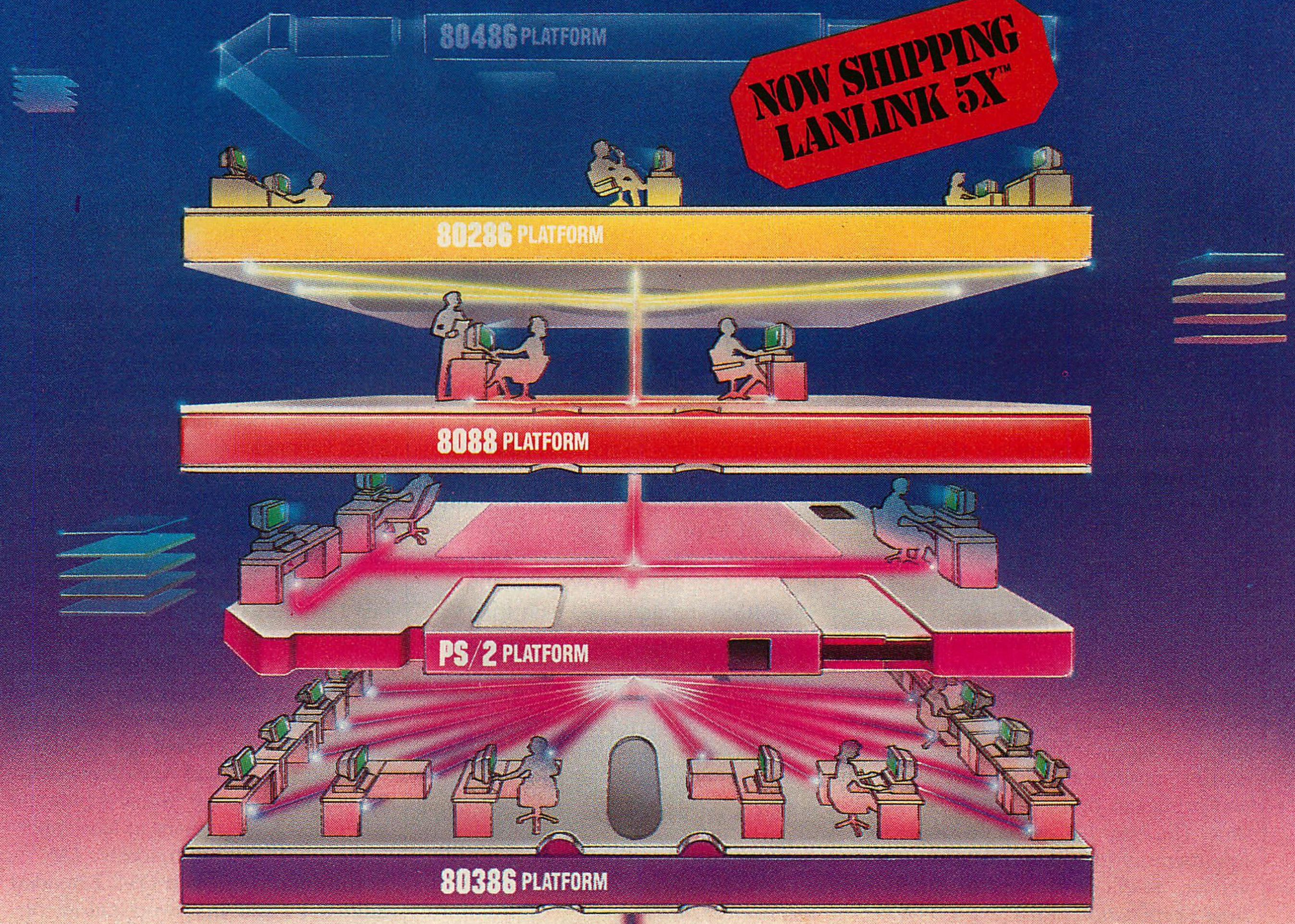
The **OS/2 Local Area Network Server version 1.0** licensed program, scheduled for shipment in November 1988, provides comprehensive LAN support to interconnected OS/2 Extended Edition 1.1 and DOS workstations on the IBM Token-Ring and PC networks. One-time license charge, $995.

The **IBM Local Area Network Program version 1.3**, a small licensed program that allows users to share computer resources in small businesses or departments among DOS workstations connected on the IBM Token-Ring and PC Network, will be available in July 1988. One-time license charge, $255; upgrade from 1.2, $90.

The **IBM 3270 Workstation Program (WSP) version 1.1** allows DOS 3.3 users to get information from a maximum of four concurrent host sessions, and up to two notepad sessions. One-time license charge, $495.

Now available from IBM are the following products: **Operating System/2 Graphics Development Toolkit**, $1,100; **Operating System/2 Programmer Toolkit 1.0**, $750; **Operating System/2 Technical Reference 1.0**, $200. Compilers for OS/2 include: the **IBM C Compiler C/2**, $525; **COBOL/2**, $900; **FORTRAN/2**, $595; **Macro Assembler/2**, $225;

THE COMPLETE NETWORK SOLUTION IS NETWORK BOARD FREE.



Making the right connections. The decision is yours. Now that most companies have multiple levels of computing power, you need more than just a short-term answer to your networking demands.

You know what you need...DOS program compatibility, multi-tasking, expandability, file/record locking with password-protected security, remote access, and ease of use. In short, you need LANLink™...the complete networking solution.

Network Board Free...Network Operating System Complete. In 1985, LANLink™ was the first network to be free of network boards. All of the network logic was on Server and Satellite diskettes. To this day, all it takes to set up a LANLink™ network is inexpensive cable, network software, and the very same communications ports most PCs & PS/2s already have.

And now, LANLink™ comes with its own network operating system...PC-MOS/386™ So you're no longer dependent on a system designed for single users and stand-alone computers.

The First Network You Buy...The Last Network You'll Need. Designed to take full advantage of the newest 80386 machines, LANLink™ provides a true multi-user system which supports the complete line of PCs, PS/2s, and PC-compatibles.

It lets you expand as your office networking needs grow. Each user gets multi-tasking capabilities, and you can network different types of computers. If desired, you can have multiple servers. And with the terminal support upgrade, you're able to use terminals, or PCs, as satellites in multi-user "work groups."

DOS Program Compatibility...Complete Connectivity. dBASE III, WordPerfect, Lotus 1-2-3, and Symphony, are among the thousands of DOS-programs that are LANLink™ compatible. The network enables security-cleared users to access and share everything from programs and databases to high-speed laser printers and large-capacity hard disks. R-LAN™ or Remote-LAN, gives you the ability to access the LANLink™ system, via modem, whether you're across the street or across the country.

A Platform for YOUR Future. The choice is clear. You can pay more than you want, for a stack of network boards. You can get less than you need with a CheapLAN—that's file transfer software which masquerades as a network. Or, you can get LANLink™ And install a SOLUTION that will take you far into the future. Its price of $495 includes a server and a satellite module plus the network operating system. For complete details and the authorized dealer nearest you, call The Software Link TODAY at the toll-free number listed below.

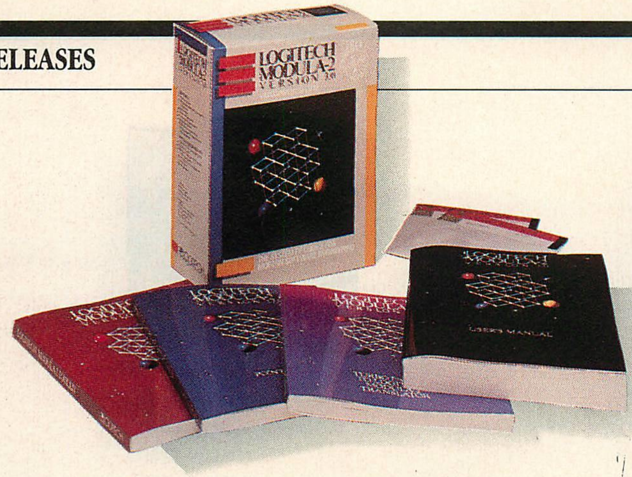
CALL: 800/451-LINK

In Georgia:	International/OEM Sales:	Resellers/VARs:	Canada:
404/441-2580	404/263-1006	404/448-5465	800/387-0453

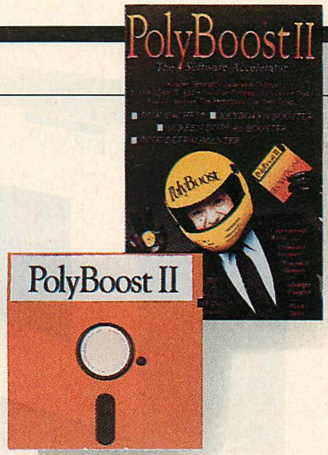
3577 Parkway Lane, Atlanta, GA 30092 Telex 4996147 SWLINK FAX 404/263-6474

CIRCLE NO. 196
ON READER SERVICE CARD

LANLink™
THE SOFTWARE LINK
Dealer Inquiries Invited



Version 3 of Modula-2 from LOGITECH

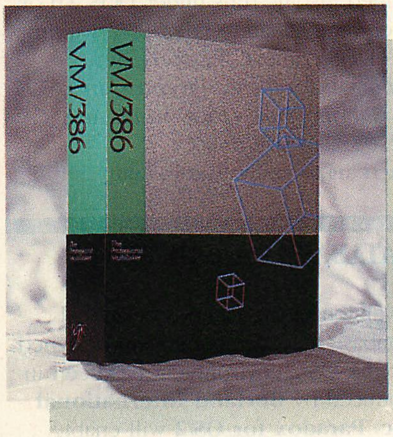


PolyBoost II from Polytron Corp.

BASIC Compiler/2, $595; and **Pascal Compiler/2**, $595. Scheduled for shipment in October 1988 is **Operating System/2 Programmer's Toolkit 1.1**, $750; and **Operating System/2 Technical Reference 1.1**, $200.
IBM Corporation, Information Systems Group, 900 King Street, Rye Brook, NY 10573; 800/426-2468; for nearest dealer, 800/447-4700
CIRCLE 318 ON READER SERVICE CARD

A software accelerator and disk defragmenter has been introduced by **Polytron Corporation**. **PolyBoost II**, is a set of 10 separately installable utilities that increase performance and power in software acceleration, hard-disk defragmentation, and memory management. Without the need for hardware, chips, or modification of application programs, PolyBoost II addresses the three areas of I/O: disk access, screen display, and keyboard input. If PolyBoost users already have an accelerator card with a faster processor, the software still will be accelerated because PolyBoost II speeds the flow of information to and from the processor. The user can set up a disk cache in RAM that will accelerate applications programs by reducing the number of times the program has to access a diskette or hard-disk drive for data. Disk caches can be located in either conventional, expanded, or extended memory and they can accommodate any size of hard disk. $79.95; upgrade from PolyBoost, $30.00
Polytron Corporation, 1815 N.W. 169th Place, Suite 2110, Beaverton, OR 97006; 800/547-4000; 503/645-1150
CIRCLE 322 ON READER SERVICE CARD

An advanced multitasking control program, **VM/386,** has been announced by **IGC, Inc.** VM/386 uses the virtual 8086 mode of the 80386 to create a series of "guest" virtual machines, each of which thinks it has exclusive access to all of the resources of the real computer. Multitasking is obtained by running a different application in each of the virtual machines; each has its own operating system and may have terminate-and-stay-resident (TSR) programs. Individual virtual machines can be customized to maximize the overall performance of 386 systems. VM/386 is compatible with existing DOS software. It also offers full EGA support to allow two EGA applications to run perfectly in the foreground and background. VM/386's System Resource Monitor allows users to specify and/or redirect

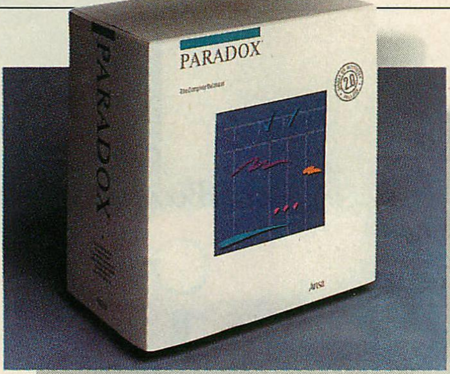


IGC's VM/386 advanced multitasking control program

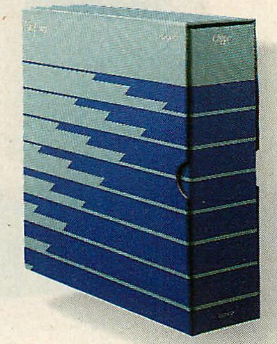
processing power assigned to tasks, and tasks can be prioritized so as to allow time-critical tasks to service external interrupts first. $245.
IGC, Inc., 4800 Great America Parkway, Santa Clara, CA 95054; 408/986-8373
CIRCLE 321 ON READER SERVICE CARD

A substantially improved version of the Modula-2 programming language development system by **LOGITECH** has been released. **LOGITECH Modula-2 version 3** offers the standard features of the Modula-2 programming language, such as its modular structure, separate compilation, and built-in multitasking. New features include its all-new user interface and standard .OBJ files that allow users to link files written in other languages (including Pascal, C and Assembler), enhancing the reusability of code and opening the system to a wider range of applications. A mouse-based user interface in the editor and the debuggers features windowing and color support and a point-and-select menu system. Version 3.0 consists of a **Compiler Pack** and **Toolkit**, both of which may be purchased separately or combined as the **LOGITECH Modula-2 Version 3.0 Development System**. The Compiler Pack consists of the compiler, linkable library, post-mortem debugger, and version 1.5 of the Point Editor, featuring syntax checking and specific support for writing Modula-2 programs. The Toolkit includes library sources, linker, runtime debugger, make, decoder, cross reference, and formatter. Modula-2 Development System, $249; purchased separately—Compiler Pack, $99 and Toolkit, $169; Window Package, $49; upgrade for version 2.0 users, $89.
LOGITECH, 6505 Kaiser Drive, Fremont, CA 94555; 800/231-7717; 415/795-8500
CIRCLE 320 ON READER SERVICE CARD

A comprehensive COBOL applications development environment from **Austec, Inc.** integrates several software development tools in a single package. **RM/MASTER COBOL**, includes a high-level ANSI 74 COBOL compiler based on the company's ACECOBOL technology, full-screen editor, screen painter, report program generator, and data dictionary. A menu-development system and bridge that provides a standard interface between applications software and dissimilar hardware and operating systems based on Austec's ACEBRIDGE allow software applications to be ported to different machine environments without change, and data files



Borland's database management system Pardox for OS/2



Nantucket's Clipper Summer '87 dBASE compiler

can be easily exchanged between dissimilar machines. A package of optional program-generator extensions is available to enable the developer to add programs quickly for such tasks as file inquiry, file maintenance, file updating, and forms printing. DOS version, $1,350; UNIX version, $2,000 to $12,000 (depending on specifications).
Austec, Inc., San Jose Office, 1740 Technology Drive, Suite 300, San Jose, CA 95110; 408/279-5533
CIRCLE 323 ON READER SERVICE CARD

A screen-management package for both C and Pascal programmers, **PC/Forms version 1.1,** has been released by **Golden Software**. Features include a form editor with standard field-editing attributes, a structured runtime library with source code, program-development and form-test utilities, windowing, scrolling fields, access to field parameters at runtime, and no runtime royalties. Turbo Pascal version, $99.95; MS-Pascal version and C version (Borland, Microsoft, and Lattice), $149 each.
Golden Software, P.O. Box 22216, 23500 Mercantile Road, Beachwood, OH 44122; 800/338-6754
CIRCLE 325 ON READER SERVICE CARD

California Software Products, Inc. has upgraded its **CALIFORNIA 10 PAK** set of programmer's tools and its **HELPME** diagnostic utility for use with OS/2 on the IBM PS/2 and IBM-compatible PCs. CALIFORNIA 10 PAK contains 16 programs for browsing, comparing, and sorting the contents of files and memory in ways impossible with standard operating system commands. System configuration and a map of all installed memory can be displayed. A dissassembler produces source files that are ready to edit and assemble from .COM files, .EXE files, or any area of main memory. An operating shell allows users to define the operation of function keys and also to create color menus and help screens. CALIFORNIA 10 PAK runs under any version of DOS and under OS/2 in either protected mode or unprotected mode. $79.

HELPME utility software is a set of more than 300 diagnostic tests that identify system configuration (including many accessories), deviations from the IBM standard, and the current use of memory and DOS interrupts. The utility also analyzes and corrects CONFIG.SYS and AUTOEXEC.BAT files and now includes checks for statements specific to the OS/2 environment. Help screens and several other functions are included. HELPME runs under DOS 2.0 or later versions and under OS/2 within the DOS compatibility box. $99.
California Software Products, Inc., 525 N. Cabrillo Park Drive, Santa Ana, CA 92701; 714/973-0440
CIRCLE 319 ON READER SERVICE CARD

DATABASE MANAGEMENT

A version of the Paradox relational database management system designed for the OS/2 environment has been introduced by **Borland International Inc. Paradox for OS/2** will enable database applications to break through the 640KB memory barrier and fully use protected-mode extended memory. It includes the following features: increased number of tables, increased records by table (up to 2 billion); validity checking; support for date formats such as alphanumeric, numeric, data, and currency; queries; updated report facilities; import/export capabilities; multiple levels of password protection at field, table, and user levels; protection and encryption support on multiple levels; and indexing. Price is not yet available.

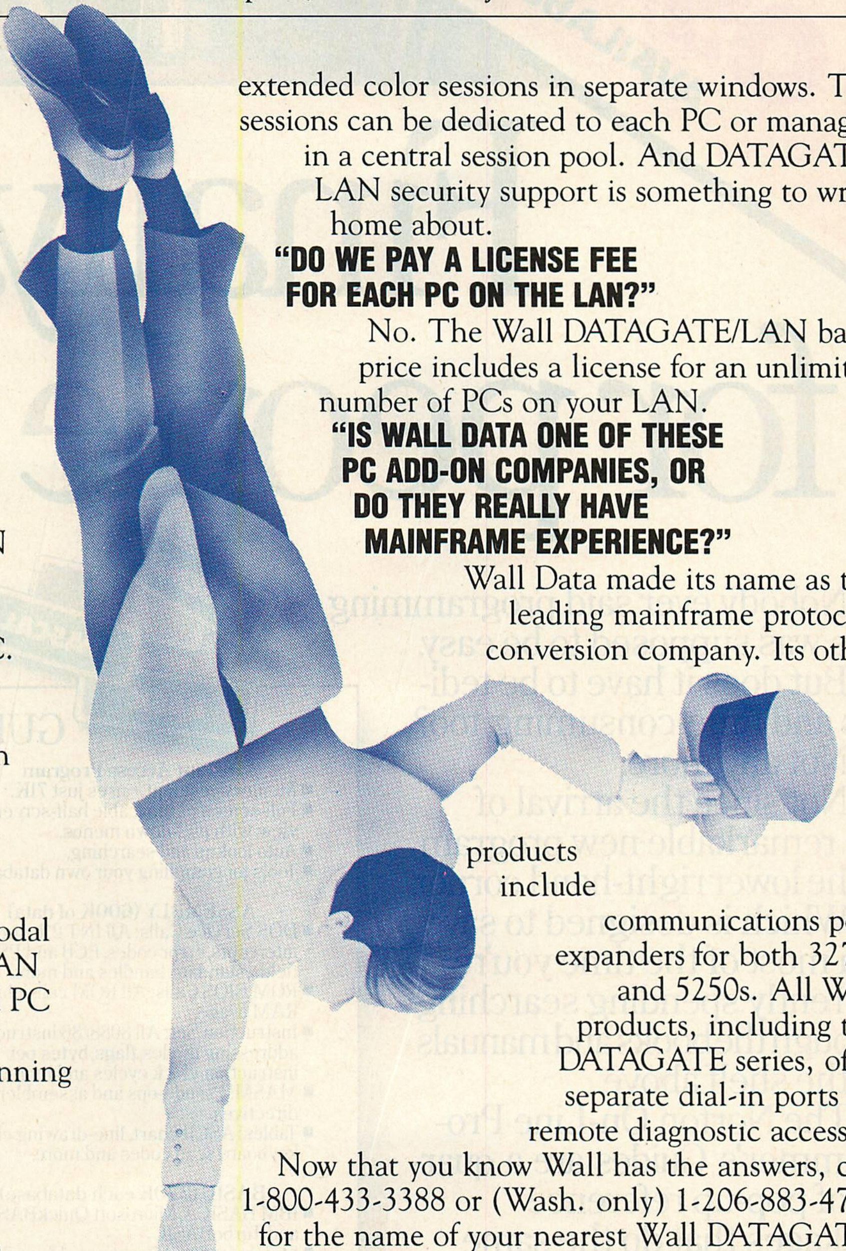
Borland also has introduced **Paradox 386**, a version that supports Intel's 80386 technology. This product breaks through the 640KB barrier using the 32-bit instruction set to speed performance of typical database operations, and using the large memory address space provided by the 386 architecture. Paradox 386 has embedded within it the 386/DOS extender manufactured by Phar Lap Software; the user can call up the program as in previous versions of Paradox. $895.
Borland International, 4585 Scotts Valley Drive, Scotts Valley, CA 95066; 408/438-8400
CIRCLE 326 ON READER SERVICE CARD

Significantly faster compilation and execution speeds are the benefits of an enhanced version of **Nantucket Corporation**'s dBASE compiler, **Clipper. Clipper Summer '87** features include optional dBASE-compatible indexes, new commands and functions, and a completely rewritten manual. The package offers expanded string-handling capabilities allowing up to 64,000 characters for faster manipulation, a command for relative SEEKing, and a browse-like DBEDIT() function. Other features include a menu-driven debugger, label and report-generator utilities, and a data structure creation and modification utility to create, browse, and edit database files. Clipper Summer '87 compiles both single- and multiuser networking applications written in dBASE or completely in Clipper, and converts them into stand-alone executable files. It offers support for up to 2,048 active memory variables, 1,024 fields in a single database record, multiple child-parent data file relations, and For/Next programming structure. $695.
Nantucket Corporation, 12555 W. Jefferson Blvd., Suite 300, Los Angeles, CA 90066; 213/390-7923
CIRCLE 328 ON READER SERVICE CARD

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

Show Off With The New Wall High-Performance LAN Gateway

Announcing the DATAGATE™/LAN from Wall Data. A high-performance LAN-to-IBM gateway that's NetBIOS compatible, can link your LAN to BSC and SNA hosts, doesn't have to reside on the server and doesn't require a dedicated PC. $1995 complete, with license for unlimited workstations.



Interested in LAN-to-IBM gateways? You'll buy the DATAGATE™/LAN from Wall Data after you ask us the following questions.

"IS THE DATAGATE/LAN COMPATIBLE WITH OUR LAN?"

Yes. The Wall DATAGATE/LAN operates with any NetBIOS-compatible LAN, such as IBM Token Ring, Novell and 3Com.

"DO WE HAVE TO DEDICATE A GATEWAY PC?"

No! The Wall DATAGATE/LAN is a co-processor that functions simultaneously with anything else you happen to be doing on your PC. Wordprocessing, spreadsheets, etc. And it won't steal cycles from your LAN database server, because it can be installed on any non-server PC.

"CAN ONE LAN BE CONNECTED TO MORE THAN ONE HOST?"

Yes. It isn't necessary to connect your host mainframes via an internodal network. The Wall DATAGATE/LAN can be connected to two hosts, and PC users can hot-key between different sessions. Both hosts can even be running different protocols: SNA and/or BISYNC.

"HOW MANY SIMULTANEOUS HOST SESSIONS CAN BE ACTIVE?"

The Wall DATAGATE/LAN can support up to 128 host sessions on your network, with up to 7 sessions per PC. This is substantially more capability than IBM or Novell offer.

"WHAT ABOUT COLOR, SECURITY & SESSION MANAGEMENT?"

Yes, yes and phenomenal. You can hot-key between MS-DOS and multiple, extended color sessions in separate windows. The sessions can be dedicated to each PC or managed in a central session pool. And DATAGATE/LAN security support is something to write home about.

"DO WE PAY A LICENSE FEE FOR EACH PC ON THE LAN?"

No. The Wall DATAGATE/LAN basic price includes a license for an unlimited number of PCs on your LAN.

"IS WALL DATA ONE OF THESE PC ADD-ON COMPANIES, OR DO THEY REALLY HAVE MAINFRAME EXPERIENCE?"

Wall Data made its name as the leading mainframe protocol-conversion company. Its other products include communications port expanders for both 3270s and 5250s. All Wall products, including the DATAGATE series, offer separate dial-in ports for remote diagnostic access.

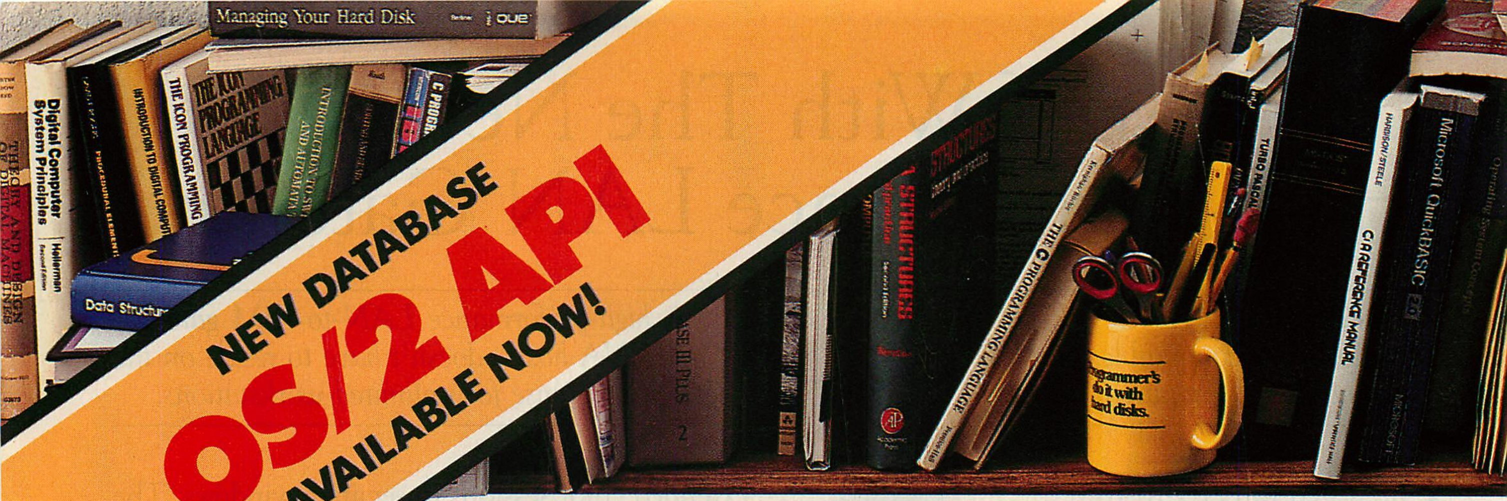
Now that you know Wall has the answers, call 1-800-433-3388 or (Wash. only) 1-206-883-4777 for the name of your nearest Wall DATAGATE/LAN reseller.

WALLDATA

Wall Data Incorporated • 17769 Northeast 78th Place
Redmond, Washington 98052-4992 • Telex 265211

CIRCLE NO. 246 ON READER SERVICE CARD

NEW DATABASE
OS/2 API
AVAILABLE NOW!



Finally. A pro for people who h

Nobody ever said programming PCs was supposed to be easy.

But does it have to be tedious and time-consuming, too?

Not any more.

Not since the arrival of the remarkable new program in the lower right-hand corner.

Which is designed to save you most of the time you're currently spending searching through the books and manuals on the shelf above.

The Norton On-Line Programmer's Guides are a quartet of pop-up reference packages that do the same things in four different languages.

Each package consists of two parts: A memory-resident instant access program. And a comprehensive, cross-referenced database crammed with just about everything you need to know to program in your favorite language.

GUIDES DATA

Instant Access Program
- Memory-resident—uses just 71K.
- Full-screen or moveable half-screen view, with pull-down menus.
- Auto lookup and searching.
- Tools for compiling your own databases.

ASSEMBLY (600K of data)
- DOS Service Calls: All INT 21h services, interrupts, error codes, FCB and PSP fields, standard handles and more.
- ROM BIOS Calls: All ROM calls plus low RAM usage.
- Instruction Set: All 8088/86 instructions, addressing modes, flags, bytes per instruction, clock cycles and more.
- MASM: Pseudo-ops and assembler directives.
- Tables: ASCII chart, line-drawing charts, keyboard scan codes and more.

BASIC (270K each database)
- IBM BASICA, Microsoft QuickBASIC and TurboBASIC.
- Statements and Functions: Describes all statements and built-in library functions.
- Tables: Line-drawing characters, ASCII chart, keyboard codes, error codes, operators, etc.

C (600K each database)
- Microsoft C and Turbo C: Describes language, including statements, operators, data types and structures.
- Library Functions: Detailed descriptions of all functions, from abort () to write ().
- Preprocessor Directives: Describes commands, usage and syntax.
- Tables: ASCII chart, line-drawing characters, keyboard codes, error codes, operators, etc.

PASCAL—Turbo (360K of data)
- Language: Describes statements, syntax, operators, data types and records.
- Library: Describes the library procedures and functions.
- Tables: ASCII chart, line-drawing characters, keyboard codes, error codes, reserved words, etc.

(If you don't believe us, you might want to take a moment or two to examine the data box you just passed.)

You can, of course, find most of this

Designed for the IBM® PC, PC-AT and DOS compatibles. Available at most software

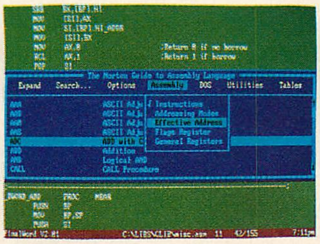
gramming tool
ate manual labor.

information in the books and manuals on our shelf.

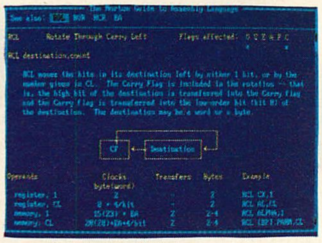
But Peter Norton—who's written a few books himself—figured you'd rather have it on your screen.

In seconds.

In either full-screen or moveable half-



A Guides reference summary screen (shown in blue) pops up on top of the program you're working on (shown in green).



Summary data expands on command into extensive detail. And you can select from a wide variety of information.

screen mode.

Popping up right next to your work. Right where you need it.

This, you're probably thinking, is precisely the kind of thinking that produced the classic Norton Utilities.

And you're right.

But even Peter Norton can't think of everything.

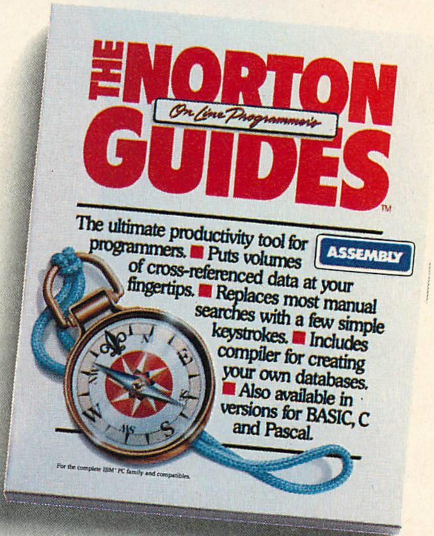
Which is why there's a built-in compiler for creating databases of your own.

And why all Guides databases are compatible with the instant access program in your original package.

So you can add more languages without spending a lot more money.

To get more information, call your dealer. Or call Peter Norton at 1-800-451-0303 Ext. 40.

And ask for some guidance.



Peter Norton
COMPUTING

dealers, or direct from Peter Norton Computing, Inc., 2210 Wilshire Blvd. #186, Santa Monica, CA 90403. 213-453-2361. Fax 213-453-6398, MCI Mail: PNCI ©1987 Peter Norton Computing

CIRCLE NO. 183 ON READER SERVICE CARD


7 or 8 bits */ int
addr; /* buffer for
/* buffer for chars being read */ int
char **argv; { int

C Contenders

Who wins? Software developers who work in C. They can choose from a strong field of veteran and rookie compilers and have at their disposable a professional set of tools and individual package enhancements.

MARTY FRANZ

Once considered the gospel only among the UNIX faithful, the C language now has ministries among DOS disciples. After only a few years, C has become the language of choice for the computer of choice.

Naturally, C presents a competitive environment for compiler vendors. Even so, most of the improvements in new releases of compilers have been incremental. Only Microsoft, with its announcement of QuickC, and Borland with Turbo C, have made radical changes to the typical C compiler for the PC. Their integrated environments are covered in detail in a separate article in this cover suite (See "Turbo and Quick Weigh In," Marty Franz, p. 72); however, the command-line versions of these two compilers are included here for comparison purposes. Other vendors have steadily improved their C compilers, but reveal no earth-shaking technological breakthroughs.

A great deal has happened in the two years since our last look at "The State of C" (William J. Hunt, January 1986, p. 82). IBM introduced the PS/2 series with the radically changed Micro Channel Architecture. A new operating environment, Microsoft Windows, has begun to gain acceptance, and another, OS/2, has just been released. Today's C developers must position themselves for these new environments and hardware, and here the vaunted portability of the C language is a real asset.

For users developing in-house software products, compatibility with Windows now (and OS/2 with or without the Windows-like Presentation Manager in the future) should be considered when choosing a C compiler. Obviously, Microsoft's compiler enjoys an enormous advantage here.

The compilers reviewed in this roundup are: C Ware DeSmet DC88 Development Package, Computer Innovations C86Plus, Datalight Optimum-C, Ecosoft Eco-C88, Lattice MS-DOS C, Manx Aztec C86, Mark Williams Let's C, MetaWare High C, and Microsoft C.

They were evaluated with particular attention to the following features:

- Support for multiple memory models. (See the sidebar "Memory Models" on page 56 for more information.) At minimum, all of these compilers provide small and large models.
- A MAKE facility for compiling and linking multiple-module programs. This uses a file of modules and their dependencies and allows the programmer to recompile and relink only the pieces of a program that have been changed since the last time the program was built.
- Availability of add-on libraries and source-level debuggers.
- Direct assembly language output from the compiler for tweaking or to permit placing code into ROM.
- Object compatibility with PC-standard MASM and LINK.
- Code generation for the 80186 and 80286. Though not used on many desktop PCs, the 80186 chip is used in many controller applications.
- Implementation of the draft ANSI language standard (see the sidebar "The ANSI Standard for C" on page 54).

PHOTOGRAPHY • MARC DAVID COHEN

This includes function prototyping and other ANSI lexical extensions, plus an ANSI-standard library.
- Support for Pascal and/or FORTRAN calling sequences, allowing object modules to be used in mixed-language applications.

The ANSI standard heralds major changes in the way C programmers write and compile software. While still officially in draft form, the standard is fairly well established. Fortunately, all compilers in this review implement some degree of compatibility with this standard. A major goal of the ANSI committee was to avoid breaking existing programs, and the goal was met in most cases. Even so, users should be aware of the differences and plan the migration of their large applications to the new ANSI standard now so they do not find themselves unprepared later.

Most of these compilers were reviewed in the 1986 round-up. In fact, Borland's Turbo C and Metaware's High C are the only completely new compilers to appear, which says a great deal about the maturity and competitiveness of this software market. Turbo C and QuickC both offer integrated environments to develop software in C. Because both offer command-line versions of the integrated environments, the products are discussed in the section on benchmarks farther along.

C CRITIQUE

The evaluation criteria and benchmark tests used in this review, with few changes, are the same used in the previous review of C compilers. One additional test for numerical accuracy has been added (see "Measuring Numerical Accuracy", Jim Roberts, January 1988, p. 142). For this evaluation, four major areas are examined:

First, basic specifications. This includes the level of language compatibility, disk and RAM used, and other basics that determine the applicability of the compiler (as summarized in table 1). ANSI C compatibility has been added as a criterion because that standard is nearing acceptance.

A look at compiler and linker features includes not only the memory models and linker formats, but the facilities provided for compiling and linking multimodule programs. This year, highly interactive programming

THE ANSI STANDARD FOR C

Like most languages, human and otherwise, C has evolved through use. The C language used on computer systems these days is not the language defined in the widely recognized standard text, *The C Programming Language* by Brian Kernighan and Dennis Ritchie (Prentice-Hall, 1978).

The authors left some areas of the language definition ambiguous. This has caused problems when programs are ported from one machine to another. In particular, function arguments, preprocessor directives, library standards, and mathematical precision have long been areas in which compiler writers have been left to implement features that could not be readily ported to other compilers.

To address these ambiguities, compiler developers and UNIX users have made their own modifications to the language. Two of the most common supersets of Kernighan and Ritchie C are the so-called "Berkeley extensions," which include enumerated data types (the **enum** declarator), void functions, and structure assignment, and the Microsoft compiler extensions, which include declarators such as **near**, **far**, and **pascal**.

Committee X3J11 of the American National Standards Institute (ANSI) formulated a standard for the C programming language. More than 130 representatives from government, academic, and industry sources have worked on the standard, now in draft form and circulating for formal comment. The changes encompass a new set of lexical constructs, a new standard library, and a new set of environmental parameters so C compilers can be used in multinational settings.

LEXICAL CHANGES

Mismatching function arguments has long been a problem in C, which allows separate compilation. The declaration of a function in one source module may not match the usage of that function in another. When this happens, a variety of unspecified conversions can occur, from lengthening of data to unintended truncation. Mismatched function arguments are a source of bugs in C programs.

The ANSI Standard for C calls for the use of *function prototypes* that specify exactly the number and type of a function's arguments. For example, in "old style" C, the declaration for **strcmp()** might be:

```
int strcmp(s1, s2)
char *s1, *s2;
{
    . . . body of function . . .
}
```

In ANSI Standard C, the prototype for this function would be

```
int strcmp(char *s1, char *s2);
```

This prototype statement (not the rest of the function's definition) would have to be included in every module that uses **strcmp( )** so that the compiler can enforce function usage.

PC compilers have included tools for assisting in this process. Some, for example, can be instructed to optionally check prototypes for validity, allowing new code to be compiled and receive the benefit of the prototype, while allowing old code to slide by until it is converted later.

For functions that use a variable number of arguments, such as **printf( )**, ellipsis put in a function's prototype conveys this fact:

```
int printf(char *format, . . . );
```

This disables argument checking after the first argument and allows arguments to be passed to the function with no conversion attempted. To indicate a function that has no arguments, the standard specifies using:

```
int f(void);
```

instead of the prestandard

```
int f();
```

The C preprocessor also benefits. No provision in "UNIX standard" C exists for anything other than an **if-else-endif** preprocessor declaration. The ANSI Standard defines an **else-if** construct that handles multiple cases. The old preprocessor directives were:

```
#ifdef TOKEN
 . . . stuff compiled if defined . . .
#endif
#ifndef TOKEN
 . . . stuff compiled if not defined . . .
#endif
```

The new preprocessor directive is:

```
#if TOKEN
 . . . stuff compiled if defined . . .
#else
 . . . stuff compiled if not defined . . .
#endif
```

which certainly resembles the syntax of the C language more closely.

environments are more common. This section now includes checkoffs for an editor, debugger, and MAKE facility (features are listed in table 2).

A quality C compiler is useless with inadequate documentation. This section reprises the previous article's documentation criteria (see table 3). This year, with inexpensive desktop publishing available, each compiler features professionally typeset or laser-printed manuals.

Last, but far from least, is a comparison of compiler performance. The standard *PC Tech Journal* benchmark suite was run. The tests are discussed in more detail below, and the results summarized in table 4.

C Ware Corporation. The DeSmet DC88 Development Package, a veteran C contender, is a full-featured compiler with a lot of added utilities for $99.

The DeSmet package comes on two 5.25-inch diskettes, with a single, standard IBM-sized manual. Installation consists of copying the contents of the diskettes to either a hard disk or a working diskette. This compact package, requiring only 256KB RAM, can be run from a single diskette. For systems with more memory, it has a RAM disk driver to boost the compiler's speed by placing more frequently used files on the RAM disk.

Along with the compiler, DeSmet DC88 includes an editor, assembler, linker, librarian, code profiler, debugger, and program listing formatter. A large-model option—for programs that use more than 64KB of code and 64KB of data—is available for an additional $79. (This is the only extra memory model offered.)

SEE, the package's fairly convenient editor, has a command structure unlike any other PC programmer's editor; however, it does offer amenities such as word wrap and macros. The C88 compiler and the BIND linker can be invoked directly from the editor. When an error occurs, the compiler returns to the editor, places the cursor on the offending line, and displays the error message. This gives DeSmet DC88 some of the flavor of an integrated environment; it also speeds program development.

The compiler does not support the full ANSI standard, although it does incorporate some of the Berkeley ex-

The **#pragma** statement has also been defined as part of the ANSI standard. This allows compiler-specific directives to be embedded in source files, such as selection of memory model or math library. If a particular **#pragma** is not supported by a compiler, it is ignored.

New declarators have been added as part of the ANSI standard. The **const** declarator makes an identifier into a constant; it can be initialized but never used as an **lvalue** (the target of an assignment). This allows the compiler to place the variable in a static or shared data area or to perform other optimizations on it.

The **volatile** declarator tells the compiler that a variable can be modified asynchronously; the compiler should not make any assumptions about it for the purposes of optimization (for example, the compiler should not assume the current value of a variable is in the register where it was last used, because another process may have modified its original location).

Both the **const** and **volatile** key words reflect C's use in realtime and multitasking systems, balanced against the need for better optimization on microcomputer systems.

Finally, the standard has defined sequences of three characters called *trigraphs* to be legal substitutes for some of C's special characters. This enables C programs to be edited and compiled on computer systems that do not support the full ASCII character set. For example, systems that do not have the { and } characters can use ??< and ??> instead. Generally, trigraphs are used only in desperation, but their inclusion indicates the nature of the issues the Committee is addressing with the new standard.

LIBRARY CHANGES

The C runtime library has been an area in which compiler implementors have had unchecked freedom. No longer. The ANSI standard created many changes to the standard C library, with the goal of allowing C programs that follows the standard to be easily ported to other systems. In particular, date and time have clearly defined routines, and functions return information on the country of the system, including that country's alphabet, collation (order of the character set), and currency symbols.

The mathematical library routines have been specified in more detail and with more rigor than on UNIX. The **#include** files **limits.h** and **float.h** specify the maxima and minima for all integral and floating-point types. A new declarator, **long double**, has been added to hold the results of math coprocessors such as the 8087 and 68881. Finally, the handling of exceptions, such as overflow and division by zero, has been standardized. All of these changes make C a stronger language for numerical data processing applications.

The following common **#include** files have now been rigorously defined by the standard: **stdio.h**, **float.h**, **limits.h**, **stddef.h**, **ctype.h**, **locale.h**, **math.h**, **setjmp.h**, **signal.h**, **stdarg.h**, **string.h**, and **time.h**.

While these files (and the functions they support) have existed in various forms on many compilers and operating systems, they are now available on all implementations supporting the standard. The functions **fseek** and **ftell** work properly on mainframe systems, where huge files may be accessed and a **long in/t** is too small to hold file offsets.

FUTURE DIRECTIONS

At the moment, most C compilers still support the old UNIX version of the language and some level of the new draft standard. As the standard edges towards completion, it will become more important to convert old-style C programs over to it so they can compile cleanly. For the time being, however, both are coexisting.

Currently, the ANSI C has no test suite, so compiler vendors are forced to track the ANSI Standard on an ad-hoc basis. This can lead to conflicts between what the compiler supports and what the standard calls for. The documentation for your favorite compiler will remain the best guide until the standard has been formally adopted and in use for awhile.

In the future, however, the ANSI Standard may be the only version compilers will support. Some compilers have included prototyping aids for functions, but the best plan is to convert your applications as you maintain them. The benefits of conversion are worth the price: with the ANSI Standard, C can finally live up to its claim of near universal portability.

—Marty Franz

tensions (see "The ANSI Standard for C" sidebar for an explanation of Berkeley extensions), such as enumerated types, **void** functions, and structure assignments. Its single-pass design means that some programs may need to be changed to run under it.

The compiler and linker have a sensible, but not overly flexible, set of options. The package has no MAKE facility, and it is best used for writing small programs. A check option generates added information in the compiler's object for the D88 debugger and code profiler. Neither the compiler nor the linker handle wild cards—a nuisance. The BIND linker uses a proprietary format, but it is faster than with Microsoft LINK; it also has a DOS LINK option. The compiler and linker can generate simple overlays.

The profiler can analyze programs compiled with the -C option to determine where the program spends its time. The D88 debugger also requires -C. It can set breakpoints, single step, display variables, unassemble instructions, and process limited expressions. While not as comprehensive as Microsoft CodeView or Mark Williams CSD, it can debug moderately sized programs.

Another strong point of DeSmet DC88 is its library, available in both 8087 and 8087-emulator versions. Along with the usual UNIX functions, the library has DOS 3.x network facilities (locking and unlocking files), PC screen and console handling functions, and interrupt call capabilities.

DeSmet has acceptable documentation, although an index would be welcome and the assembly language and runtime technical information is sparse. The documentation appears to have been recently rewritten and typeset; it includes details on every error message the utilities produce. A bulletin board system provides technical support.

The benchmarks show this compiler to be a speed demon when it comes to compiling and linking, with average execution times and module sizes. DeSmet is a good compiler for a single programmer or small group. It is compact, fast, and includes convenient utilities and a library at a low price.

Computer Innovations Inc. C86Plus would have done well in this review if the actual performance of the compiler had lived up to the promise of the documentation and features. Although it has many things in place, it does not live up to the promise.

Computer Innovations produced one of the earliest C compilers for the IBM PC. This compiler has gained a

TABLE 1: *Compiler Specifications*

	C WARE	COMPUTER INNOVATIONS	DATALIGHT	ECOSOFT
PRODUCT	DeSmet C	C86Plus	Optimum-C	Eco-C88
VERSION TESTED	3.0	1.10	3.11	4.01
PRICE	$99	$497	$139	$99.95
FEATURES				
Supported on other systems	●	○	○	○
Cross-compiler hosts	○	○	○	○
Add-on library availability	Good	Good	Good	Fair
Minimum disk space required	482KB	2MB	720KB	1MB
Minimum RAM required	192KB	512KB	180KB	256KB
Supports full language	See text	●	●	●
PC-specific functions	●	●	●	●
Assembly language interface	●	●	●	●
COMPATIBILITY				
MASM	●	●	●	●
LINK	●	●	●	●
SOURCE CODE				
Start-up sequence	●	●	●	○
Library functions	$139	●	●	○
MEMORY MODELS				
Tiny	○	○	●	○
Small	●	●	●	●
Compact	○	●	●	●
Medium	○	●	●	●
Large	$79	●	●	●
Huge	○	●[a]	●[b]	○
OTHER PROGRAMS INCLUDED				
Librarian	●	●	○	○
Assembler	●	○	○	○
Linker	●	●	●	○
Source-level debugger	●	○	○	●
MAKE facility	○	●	●	○

● = Yes ○ = No

[a] *Via pointer declaration.*
[b] *Via casting pointer to long.*
[c] *Default for large and compact models.*

MEMORY MODELS

Unlike other programming languages, C compilers allow the programmer to control the way in which memory is used for storage of variables and object code. This is done in two ways—by compiling the program using a particular memory model, and by incorporating **#pragma** statements into the source code.

Because of the PC's segmented memory architecture, program code (the instructions that make up the program), data (variables), and stack space (variables local to functions) can be allocated several ways. Limiting one or the other to 64KB allows short (two-byte) pointers to be used instead of long (four-byte) pointers, speeding up processing and making the object code more compact. The programmer selects the memory model when the code is compiled and linked. The model, in turn, determines the type of pointers used for code and data.

The common memory models used in C compilers for the PC are:

Tiny. This is generally a .COM file layout in which the code, data, and stack segments must share a single 64KB memory area. This model has been useful for small, fast utilities, but it is a dead end in the long run: under OS/2, this type of program, with its code and data in the same segment, is not supported.

Small. This is a .EXE file with up to 64KB allowed for code and 64KB

LATTICE	MANX	MARK WILLIAMS	METAWARE	MICROSOFT	BORLAND	MICROSOFT
MS-DOS C	Aztec C86	Let's C	High C	C 5.0	Turbo C	QuickC
3.2	4.0	4.0.11	1.4	5.0	1.0	1.0
$500	$499	$75	$595	$450	$99.95	$99
●	●	●	See text	●	○	○
●	●	●	See text	●	○	○
Good	Fair	Fair	Fair	Good	Good	Good
720KB	520KB	1MB	1.5MB	1.1MB	360KB	500KB
256KB	256KB#	320KB	256KB	448KB	384KB	320KB
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	○	●	●	●	●
●	●	●	●	●	●	●
●	●	●	○	●	●	○
Some	●	○	○	$150	$150	$150
●	●	●	○	○	●	○
●	●	●	●	●	●	●
●	●	○	●	●	●	●
●	●	○	●	●	●	●
●	●	●	●	●	●	●
●[c]	●[a]	○	●[a]	●	●	○
○	●	●	○	●	●	●
○	●	●	○	○	○	○
○	●	○	○	●	●	●
○	●	●	○	●	○	●
$195	●	●	○	●	●	●

All of these compilers implement the full C language, but some include utilities and support more memory models. Turbo C and QuickC are shown here for comparison only; they are discussed in "Turbo and Quick Weigh In" on page 72.

for data. In this model, the stack is allocated within the data segment.

Compact. Code is restricted to 64KB, but data can be as large as 1MB. This is useful for programs that process a lot of data in-memory. Having a larger data space available means that larger pointers to variables are needed, slowing the program somewhat.

Medium. The code can be larger than 64KB, but data must fit in 64KB. Useful for large programs that do not process a lot of data in-memory. Access to variables remains fast since short (two-byte) pointers are used.

Large. Both the code and data segments can be as large as needed. This is used for large programs that process a lot of data in-memory.

Huge. Both data and code can be as large as 1MB, but huge pointers are "normalized" so that the segment portion of the address is as large as possible, with the offset restricted to numbers between 0x0 and 0xf. Although normalizing pointers takes more time, it assures that two pointers pointing to the same address will compare properly.

In addition to memory model, in many C compilers individual functions and variables can be made **near** or **far** (or called with an interrupt) via a special **#pragma** statement. This allows C programs to interface directly with DOS and BIOS routines or programs written in other languages.

—Marty Franz

reputation over the years as one of the most reliable, offering a good performance and a host of facilities. At $497, the price is in line with the leading C compiler, Microsoft C.

The company has completely rewritten this version; it produces smaller, faster code, and is compatible with Microsoft's linker and libraries.

The C86Plus package consists of two binders and seven diskettes. An installation program creates the appropriate directories and copies the files. Optionally, the library source code, provided in compressed archive format, can also be unpacked and placed on disk. Although the extensive amount of source code forces work to be done on a hard disk, it is the compiler's greatest strength. Nearly every function in the runtime library, including the start-up code, is available in source code format. The code readability varies, but it all can be modified. The library is among the most complete, featuring a wealth of PC-specific functions including BIOS calls for screen handling and communications.

The C86Plus compiler supports the small, medium, and large memory models, and, through a clever compilation switch set-up, allows the programmer to create custom models. (However, the library is available only in these three formats without recompilation.) The compiler flags resemble those of the Microsoft product. ROM-based code can be generated with the separately purchased ROMpac.

Like the compiler's operation, the MAKE facility is similar to Microsoft's. The package does not include an editor, but it does come with a source code archiver and an object module librarian; both serve to ease the maintenance of large projects.

The first problem with the Computer Innovations package is software release control. While this article was being written, complete upgrades appeared at the *PC Tech Journal* offices roughly once a month, each fixing minuscule bugs. It is exasperating to repeatedly reinstall the compiler with each new bug fix. It was not until the last revision was installed that the compiler could run the benchmarks. Fortunately, the company's telephone support personnel are knowledgeable and friendly, and its frequent newsletters list bug fixes and performance tips.

Another problem for C86Plus is its documentation. Although typeset, the content is spotty, especially on technical topics such as the assembly language interface. References to source

code do not help much, since the code itself has few comments.

The User's Guide is written in a lighthearted style, a welcome change from some of the other manuals reviewed. But although it adequately describes the basics of the compiler's operation, it does not go much beyond that. One item in particular is worth noting. The debugger section is marked with a tab, but has no accompanying pages—presumably awaiting future release. Computer Innovations has tried to make up for the missing material with C86INFO, a program that provides on-line documentation. But C86INFO does not contain nearly enough information, especially when describing topics such as CAROLE, the linker program, and its options.

The first version tested (1.09) erroneously detected an 8087 in all the benchmark machines (including a stock IBM PC), and generated a floating-point exception when the programs were run. A second version (1.09E) fixed the floating-point problem but had difficulty generating code to reference cast structure members, such as are found in ELTIME.C (from the benchmark suite). Once version 1.10 of the compiler was delivered, the benchmarks could be run.

In short, C86Plus was beset with quality control problems, with only a library in source code to recommend it. Computer Innovations must produce a stable, bug-free release of its product before it can once again be called a strong contender in the C marketplace.

Datalight Inc. Optimum-C is shipped on four diskettes, with a standard IBM-sized manual. The package includes the compiler, utilities such as MAKE, libraries, and source code to most of the library, start-up routines, and utilities.

Datalight C resembles Lattice C or Microsoft C in operation. An installation program copies the pertinent files onto a hard disk or diskette. The driver offers a variety of switches, including model selection, floating-point or emulation libraries, and assembly language output. More importantly, an optional optimizer pass can be run, with flags controlling optimization: constant folding, common subexpression removal, dead code removal, and others. Even so, the optimizer is not well documented in the manuals and requires a trial-and-error approach. The individual optimizations can be controlled independently, allowing an application to be "fine-tuned."

Datalight C supports the full Kernighan and Ritchie standard and the Berkeley extensions, but only part of the draft ANSI standard: it does not support Pascal calling sequences. The library is distinguished only by the source code available for most of it. It includes a few Microsoft mouse, sound, and PC display functions. Many add-on software libraries are available for this compiler, because it so closely emulates the Lattice product. (A compilation option permits exact emulation of the Lattice calling sequence.)

The documentation for Datalight C is barely adequate for technical users, with some outright errors. For example, the installation procedure does not correspond to what is shipped on diskette. In fact, it is extremely difficult to determine exactly what has been shipped with the compiler and what

TABLE 2: *Compiler Features*

	C WARE	COMPUTER INNOVATIONS	DATALIGHT	ECOSOFT
PRODUCT	DeSmet C	C86Plus	Optimum-C	Eco-C88
COMPILER OPERATION				
Single-step compile command	●	●	●	●
Single-step compile and link	●	●	●	●
Accepts list of files	●	●	●	●
Accepts wild cards	○	●	●	●
Lists preprocessor output	○	●	●	○
Lists assembler output	●	●	●	●
Line numbers in error messages	●	●	●	●
Header file search list	●	●	●	●
Flexible disk file layout	●	●	●	●
C LANGUAGE EXTENSIONS				
Embedded assembly language	●	○	○	○
Draft ANSI standard	○	●	●	●
Alternate calling convention	○	●	○	○
Interrupt function declaration	○	●[b]	●[b]	○
LIBRARY EXTENSIONS				
Math functions (sqrt, exp, etc.)	●	●	●	●
Unbuffered file I/O	●	●	●	●
Keyboard input (low-level)	●	●	●	●
PC screen output	●	●	●	●
PC graphics	●	○	○	○
Execute programs/DOS	●	●	●	●
DOS services	●	●	●	●
PC-specific funtions (other)	●	●	●	●
UNIX-compatible functions	●	●	●	●
FILE I/O				
Redirection	●	●	●	●
Full path names	●	●	●	●
DOS 3.x file sharing	●	●	○	○
DOS 3.x record locking	●	●	○	○
ASCII/binary mode	●	●	●	●
MEMORY USAGE				
Overlays	●	●	●	●
Default sack size	●	●	●	○
Stack size can be set	●	●	●	○
Stack overflow checking	○	●	●	○
8086 FAMILY SUPPORT				
Byte/word alignment	○	●	●	○
80186/286 support	○	●	○	○
8087/287 support	●	●	●	●
Automatic sensing	○	●	○	●
ROM support	○	$250	●	○

[a] *Compiler does not support multiple file names; the MAKE facility does.*
[b] *Implement via function package.*
[c] *Native overlay facility.*

LATTICE	MANX	MARK WILLIAMS	METAWARE	MICROSOFT	BORLAND	MICROSOFT
MS-DOS C	Aztec	Let's C	High C	C 5.0	Turbo C	QuickC
●	●	●	●	●	●	●
○	●	●	○	●	●	●
●	○	●	○	●	●[a]	●
●	○	●	○	●	●[a]	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
○	●	○	○	○	●	○
●	See text	●	●	●	●	●
●	○	●	●	●	●	●
○	○	○	○	●	●	●
●	●	●	●	●	●	●
●	●	●	See text	●	●	●
●	●	●	●	●	●	●
○	●	●	○	○	●	●
○	●	○	○	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	○	○	●	●	●	●
●	○	○	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	○	●	●	●	●	●
●	●	●	●	●	●	●
●	○	●	●	●	●	●
●	●	●	●	●	●	●
●	○	●	●	●	●	●
○	●	●	●	●	●	○

These compilers are competitive in feature content. The standouts provide better PC-specific features and 8087 support. Features specific to the Turbo C and QuickC integrated environments are compared in the table on page 74.

has not, because no contents list is included. In general, the manual is choppy and poorly organized.

However, the compiler's performance compensates for the weak documentation. Datalight C was among the best in program size and execution times. Datalight had particularly good times for file I/O using **getc** and **putc**, a bright spot for those using C to write small utility programs. Oddly, running the optimizer in various switch combinations did not improve performance noticeably over the unoptimized object code for most of the benchmarks. Nevertheless, with its low price ($139), Datalight is a good compiler for smaller operations.

Ecosoft Inc. Eco-C88 has been known as a low-end product limited to small model programs, but version 4.0, released in time for this review, supports more memory models and the draft ANSI standard. It also includes a full-screen editor and a debugger, making it well worth consideration, especially at the low price of $99.95.

Eco-C88 is packaged on seven diskettes, with a standard IBM-sized manual. Installing the compiler calls for running a program that prompts for directory names. Like other multimodel compilers, a hard disk is necessary.

The compiler's operation is conventional: a driver program compiles and links all the files on the command line. Optionally, you can use CED, which is a combination shell and text editor. The shell part of CED resembles Turbo Pascal 3.0 in appearance and operation, but it is clumsy to use—you still have to provide parameters to the compiler and/or linker.

A mini-MAKE option determines which files on the command line have been changed and compiles or links just those; otherwise the compiler has no MAKE. The editor part of CED uses the MicroPro WordStar editing commands for basic operations, and adds some features of its own, such as the ability to edit several files at once.

One helpful feature of Eco-C88 is its error reporting, which is very complete and includes a page reference to the *The C Programming Guide* (Purdum, Que Corporation, 1986). Various intensities of error checking can be selected, ranging from very strict to extremely forgiving. At the strictest level, it allows no implied conversions: every data type in the program must be cast. This emphasis on error checking and reporting is sure to be especially helpful to novice C programmers, but also can be useful to experienced programmers intent on creating portable code.

Another nice feature is the library, which includes PC-specific screen functions, "memory files" (buffers read and written as if they were files but residing in memory for speed), sorting and searching functions, and DOS directory handling. Source code to the library is not made available.

Eco-C88's documentation has improved over previous versions, but it is still skimpy compared with the best of the others. The performance of the compiler is average. For $99.95 though, Eco-C88 is a competent compiler with a decided orientation to the novice.

Lattice Inc. Along with Computer Innovations, Lattice had one of the earliest C compilers for the PC with its MS-DOS C Compiler. A few years ago,

Microsoft was selling Lattice's compiler for a time, before writing its own. Lattice enjoys a reputation for having a reliable, robust compiler. In addition, Lattice compilers run on Amiga and Macintosh, so portability is a strong point. To top it off, this version now supports the draft ANSI standard for C.

Lattice is the only compiler that is shipped in both 5.25-inch and 3.5-inch formats. An installation program copies the compiler, model libraries, utilities, and source code to the directories. A second, large-table version of the compiler can handle extremely large source files. Two typeset, hard-covered, spiral-bound manuals carry the documentation. But although they are strong in binding, they are weak in content when it comes to specifying the assembly language interface and the programming environment.

The Lattice compiler is easy to use, yet retains a great deal of flexibility. A MAKE facility is available separately, as is a screen editor. The package includes an object librarian and a disassembler, plus **pr**, a source code formatting utility. The library is complete and includes some useful DOS 3.x network functions. The only major omission is that programs cannot use mixed memory models: for example, **near** and **far** functions cannot be mixed.

As the benchmark results show, Lattice performs acceptably, although it has some quirks, such as the fact that its large model string copying uses register variables, which is incredibly slow. Nonetheless, it produces reasonable object code for most applications.

The Lattice C compiler has a good library, produces acceptable code, comes with fair documentation, and is complemented by good technical support. (Support from Lattice is available through BYTE Information Exchange and a private bulletin board, as well as technical support personnel.) This compiler has aged well, and is worth considering for general development.

Manx Software Systems Inc. With Aztec C86, the emphasis is on portability among systems: Manx has C compilers for the PC, the Macintosh, UNIX, CP/M, Apple II, even the Commodore 64. All Aztec compilers, regardless of the system upon which they run, operate the same way and use the same tools, down to the editor and linker.

Aztec C86 for the PC comes in three versions: the Professional System, at $199; the Developer's System, at $299; and the full-blown Commercial System, at $499. This review applies to the latter two, which include all the utilities and memory models plus the ability to generate ROM-based code.

Aztec C86 is shipped on five diskettes and comes with a giant, boxed, standard IBM-sized manual. Installation, consisting of the standard copying onto a hard disk or diskette, is poorly documented. In fact, the manual is organized in such a way as to make assembly of the various compiler chunks easy for the vendor, but difficult for the actual user. For example, information about the compiler driver program is contained in the section on the version 4.0 update, rather than the section on the compiler itself.

In the ponderous Aztec C documentation, the only bright spot is the section on version 4.0's adherence to the ANSI draft standard, which is con-

TABLE 3: ***Documentation Quality***

	C WARE	COMPUTER INNOVATIONS	DATALIGHT	ECOSOFT
PRODUCT	DeSmet	C86Plus	Optimum-C	Eco-C88
INSTALLATION				
Packing list	●	○	●	●
File inventory	●	○	●	●
Key files described	●	○	●	●
Quick step-by-step procedure	○	○	●	●
Instructions for diskette and hard disk	●	○	●	●
List changes from last version	●	●	●	○
Set-up assumptions described	●	○	●	●
OPERATIONS EXPLAINED				
Compile options	●	●	●	●
Compiler error messages	●	●	●	●
Linking C programs	●	●	●	○
Runtime error messages	●	●	●	●
Runtime options	●	●	●	○
LANGUAGE/LIBRARY SPECIFICATIONS				
Deviations from Kernighan and Ritchie documented	Good	Good	Fair	Fair
Data type representation	Fair	Good	Fair	Fair
Memory models and memory layout	Fair	Fair	Fair	Fair
DOS and PC-specific features	Good	Fair	Fair	Good
ASSEMBLY LANGUAGE INTERFACE				
Segment, group, class specification	Poor	Fair	Fair	Fair
Standard prologue, epilogue	Some	Fair	Fair	Fair
Instruction formats for arguments	Some	Fair	Fair	Fair
Return value conventions	Good	Fair	Fair	Fair
Complete examples	Fair	Fair	Fair	Fair
FILE I/O				
Redirection explained	Fair	Good	Fair	Good
Console I/O explained	Fair	Good	Fair	Good
Device I/O explained	Fair	Good	Fair	Good
Buffered versus unbuffered modes	Fair	Good	Fair	Fair
ASCII versus binary modes	Fair	Good	Fair	Fair
LIBRARY DOCUMENTATION				
Average lines per function	12	30	7	30
Cross-reference information	Fair	Good	Fair	Fair
Functions in table of contents	●	●	●	○
Examples of use	●	●	●	●
MANUAL ORGANIZATION				
Detailed table of contents	●	●	●	○
Index with functional entries	●	●	●	●
Order of function documentation	Sections	Alpha.	Alpha.	Alpha.
Overall rating	Fair	Fair	Fair	Fair

● = *Yes* ○ = *No*

LATTICE	MANX	MARK WILLIAMS	METAWARE	MICROSOFT	BORLAND	MICROSOFT
MS-DOS C	Aztec	Let's C	High C	C 5.0	Turbo C	QuickC
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	○	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	○	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
Good	Fair	Fair	Good	Fair	Good	Fair
Good	Good	Good	Good	Good	Good	Good
Good	Good	Fair	Good	Good	Good	Good
Good	Fair	Fair	Poor	Good	Good	Good
Fair	Good	Fair	Fair	Good	Good	Good
Fair	Good	Fair	Fair	Good	Good	Good
Fair	Good	Fair	Fair	Good	Good	Good
Fair	Good	Fair	Fair	Good	Good	Good
Poor	Good	Fair	Fair	Good	Good	Good
Good	Fair	Good	Poor	Good	Good	Good
Good	Fair	Good	Poor	Good	Good	Good
Good	Fair	Good	Poor	Good	Good	Good
Good	Fair	Good	Poor	Good	Good	Good
Good	Poor	Good	Fair	Good	Good	Good
50	35	20	20	50	50	50
Good	Good	Good	Fair	Good	Good	Good
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
●	●	●	●	●	●	●
Alpha. Fair	Sections Fair	Alpha. Fair to good	Alpha. Poor to fair	Alpha. Good	Alpha. Good	Alpha. Good

These products and the language they compile are complex; good documentation is required to expedite development and avert user frustration.

cise and useful. But documentation aside, the balance of Aztec C is robust.

The compiler now supports most of the draft ANSI standard with a compilation switch and all memory models except huge. The source code to the libraries is included in the Commercial system in source archive format. The compiler driver allows mixed models to be compiled and linked, a real advantage for serious developers. Most major useful UNIX utilities, such as **grep** and **diff** are included, as is an editor called Z, which mimics the UNIX **vi** editor. It even has a source debugger, SDB, and an assembler.

One drawback with porting C is the difficulty in taking advantage of the specific attributes of each machine. Fortunately, Aztec C includes many PC-specific functions, including the ability to do screen graphics with BASIC-style **point**, **palette**, and **line** functions. The Aztec C86 library is one of the most complete of the group.

This compiler turns in a good performance, even in compilation and link times, an area in which Aztec has been slow in previous versions. Its numerical accuracy also was good.

Aztec C is a comprehensive compiler, worth the attention of serious developers, especially those who need to support multiple versions of their software across different systems. The documentation and overall packaging could be improved: Aztec should go with a single, full-blown version of its compiler and drop the price/bundling structure, rewriting its documentation specifically for this version. The company would end up with something people would gladly pay $499 to use.

Mark Williams Company. Let's C started out as a hobbyist version of Mark Williams' more powerful $495 C Programming System, which has been discontinued. Now Mark Williams has enhanced Let's C with large model support, a development shell, a MAKE facility, 8087 support, and CSD, a source code debugger.

In operation, Let's C sits squarely between completely command-driven compilers (such as Lattice and Aztec) and completely integrated environments such as Turbo C. A resident shell dispatches the editor, compiler, and MAKE facility and remembers the options from the previous invocation. The shell features slick scrolling menus and text boxes but suffers from annoying flicker on some color displays.

It is packaged as a single perfect-bound paperback manual and four diskettes. When it includes CSD, the package also has another slim paperback manual and another diskette. An installation program prompts for the directory names to hold the executables, libraries, and include files, and then copies the contents of the compiler's diskettes to them.

Along with the compiler, Let's C has a development shell, an assembler, and a set of UNIX-like utilities, including **pr**, **cmp**, and **egrep**. The assembler is not source compatible with MASM, so it is mostly appropriate for rewriting C functions in assembly language for better performance.

The program editor included with Let's C is the MicroEMACS editor, modified to run under the new shell. Because Mark Williams sells COHERENT, a UNIX-compatible operating system,

you would expect MicroEMACS to be UNIX-flavored, and you would be right: control keys do everything—color, PC-specific features (boxes, function keys), and status information is kept to a minimum. MicroEMACS includes the ability to edit multiple buffers and cut and paste between them. It also offers macros and word wrap.

The rest of the environment works fairly well. Building compiler command lines consists of using the shell's point-and-shoot option selection. The compiler supports the Kernighan and Ritchie standard, plus the Berkeley extensions (**void** and **enum**), as well as the draft ANSI standard. The linker, which is now LINK and not Mark Williams' proprietary linker, is automatically invoked from the compiler without fuss. A number of options exist for controlling the level of compiler warnings and errors generated, and a particularly useful option enforces strict adherence to Kernighan and Ritchie. The compiler will now generate 80186, 80286, and ROM-based code.

The Let's C documentation is nicely designed and typeset, and its single manual convenient. After the tutorial sections, its organization is unusual: everything, from library routines and include files to language terms and utilities, is described in a single alphabetic reference section called the Lexicon. This saves search time, especially for novice programmers. The only complaint is that the information is packed in a small space, and it would benefit from more examples.

The Let's C library includes UNIX and ANSI standard functions. PC support includes BIOS and DOS access routines, but little else specific to the PC. Programs can be compiled with either an 8087-only or 8087-sensing library. Let's C performance fell in the middle of the pack, with the exception of the read/write benchmarks, which were among the best of the group.

When coupled with CSD, the Let's C shell becomes a powerful, nearly integrated workbench for C programming. Like the other debuggers reviewed here, CSD is good for small to medium programs, getting slower as the number of source lines increases (although the disk cache the shell uses helps out a lot). Its user interface makes it the easiest to use of the bunch. Its drawbacks include its screen flicker and its inability to debug library or assembly language routines.

Let's C remains a fairly powerful C development environment, with a number of features that will make UNIX and novice C programmers alike feel comfortable on the PC

TABLE 4: *Benchmark Results*

	C WARE	COMPUTER INNOVATIONS	DATALIGHT	ECOSOFT
PRODUCT	DeSmet	C86Plus	Optimum-C	Eco-C88
COMPILE TIMES (hard disk)				
60-line file	4.9	25.4	5.3	9.9
150-line file	6.4	70.8	6.9	15.5
500-line file	9.5	119.9	9.6	21.5
LINK TIMES (hard disk)				
1 object file	6.7	27.1	6.6	10.7
6 object files	8.9	30.6	7.4	12.2
PROGRAM SIZES (bytes)				
Eratosthenes Sieve	12,800	12,624	12,906	13,824
Pentathlon	14,336	35,108	14,232	15,704
EXECUTION TIMES [a] (small/large model)				
Function calls (Fibonacci)	10.4/9.2	8.1/8.1	7.5/8.5	8.1/8.6
Integer arithmetic	13.0/9.9	8.5/8.6	7.5/7.5	10.0/10.0
Long arithmetic	46.9/41.0	23.0/23.4	38.6/39.2	77.1/85.0
Subscripts	12.5/11.4	7.9/11.3	9.3/11.1	10.8/11.8
Pointer use	15.3/18.5	12.2/19.2	13.3/15.3	14.4/17.5
With register variables	76.8/91.3	51.3/96.2	66.2/76.7	70.2/87.5
Eratosthenes Sieve	9.5/7.7	5.8/5.8	6.9/6.7	7.8/9.4
With register variables	47.5/38.5	22.4/22.7	32.9/34.4	37.8/47.0
FILE I/O				
Read/write [b]				
Diskette to diskette	8.7/7.6	4.0/6.5	8.9/9.0	6.0/5.6
Hard disk to hard disk	1.8/2.3	0.9/0.9	1.1/1.2	0.8/0.8
Getc/putc [c]				
Diskette to diskette	73.1/126.5	22.47/88.1	15.2/110.6	111.9/65.6
Hard disk to hard disk	25.9/29.3	12.91/13.9	6.3/14.0	17.1/18.4
FLOATING-POINT OPERATIONS (Software/80x87 chip)				
Add/multiply (dot product) [a]	17.5/4.2	41.0/2.8	30.3/11.8	24.5/10.9
Exponential/logarithmic [d]	15.9/0.9	81.2/1.3	22.0/1.3	27.1/6.0
Sin/tan (trigonometric) [d]	18.3/6.1	42.5/1.5	22.5/1.3	51.3/6.8
NUMERICAL ERROR RATING [e]	1.5	1.0	1.1	1.0

All times are in seconds.

[a] 20 iterations. [b] 1 iteration. [c] 2 iterations. [d] 10 iterations. [e] Smaller values indicate higher numeric accuracy. [f] Compiler bug; see text.

MetaWare Inc. At a time when some C compiler vendors have downsized their products, MetaWare's relatively recent addition to the market, High C, is an exception. (But, with a number of proven compilers available, do we really need another expensive—$495, batch-style C compiler for the IBM PC?) Yes, when it's MetaWare offering High C, a genuine alternative with some unique features to appeal to professional product developers on the PC.

MetaWare is no stranger to the compiler market. In addition to High C, the company markets many other development tools, including Professional Pascal, and cross compilers for XENIX, VAX VMS, plus the Motorola 68000 and NS 32000 chip families under UNIX. Code generator support of native 386 code has made MetaWare products very popular for use with 386 DOS extenders such as Phar Lap's RUN386.

The High C compiler comes on seven diskettes with a thick loose-leaf manual in a slipcase. Installation is done with a program that creates all the necessary subdirectories and copies the various libraries and modules to them. With the libraries for all memory models taking up close to 2MB, a hard disk is definitely required. The installation process is well documented and includes several demo programs that can be run to verify that the compiler is functioning properly.

LATTICE	MANX	MARK WILLIAMS	METAWARE	MICROSOFT	BORLAND	MICROSOFT
MS-DOS C	Aztec	Let's C	High C	C 5.0	Turbo C	QuickC
13.1	14.0	7.1	38.9	12.9	4.8	6.9
22.5	20.9	11.7	52.2	23.2	6.5	9.0
37.4	29.8	17.7	66.7	34.1	8.9	11.7
15.1	6.4	9.8	13.2	7.0	7.0	7.3
23.6	11.3	11.8	16.4	10.4	9.2	10.5
10,784	9,326	12,656	23,760	10,341	8,294	10,389
18,688	10,620	14,160	25,856	26,188	23,780	26,588
7.8/8.3	8.1/8.6	9.7/10.6	7.6/9.7	7.9/8.2	6.0/7.8	7.9/8.2
7.9/7.9	9.2/9.2	9.7/11.1	7.5/7.5	7.6/7.6	7.0/7.0	10.0/9.89
23.3/24.3	23.8/24.2	36.0/38.4	27.8/29.0	26.0/26.8	29.2/29.0	27.9/29.0
9.6/33.2	9.0/10.4	10.3/12.8	9.2/10.4	7.0/7.9	8.8/9.7	9.7/12.3
12.3/57.9	12.7/15.3	13.0/15.3	12.8/15.3	13.0/15.3	13.2/15.3	15.4/20.4
61.8/290.6	38.7/76.6	33.3/76.8	63.7/76.4	28.0/76.7	38.5/76.5	54.5/102.2
7.9/8.2	7.6/7.6	7.6/9.7	6.9/6.9	7.5/7.4	4.8/5.1	8.7/8.68
40.8/39.6	29.7/30.3	30.2/40.5	34.5/33.6	28.8/28.8	32.1/32.5	36.0/36.0
6.3/6.3	8.7/7.6	5.3/5.3	2.6/2.6	2.8/2.8	17.1/16.3	2.5/2.6
0.9/1.1	3.0/3.0	1.7/1.7	0.3/0.3	0.7/0.7	2.3/3.0	0.6/0.6
63.1/63.1	40.1/66.2	8.3/9.6	82.1/105.0	104.5/99.1	63.5/106.9	104.5/102.5
15.2/21.0	10.8/13.4	10.4/12.4	14.4/24.6	16.8/17.5	11.7/13.2	18.6/19.2
19.3/13.7	13.3/5.4	20.5/3.6	16.6/3.9	13.7/3.4	23.2/3.1	21.9/3.13
15.2/1.4	17.9/5.3	21.0/0.9	20.0/1.0	12.8/1.3	7.4/1.0	13.9/1.2
18.2/2.0	20.3/6.2	24.1/1.1	18.7/1.2	13.2/1.4	8.4/1.2	14.6/1.3
0.8	0.9	0.8	0.8	0.8	0.9	3.7[e]

The benchmarks show many close contenders in compile and execution times. Tests for QuickC and Turbo C used the command-line versions of those compilers; see page 80 for results of benchmarks in their integrated environments.

Along with the compiler and libraries, High C includes seven extra UNIX-like utilities, including **mv** (move) and **fgrep** (a pattern finder). Although these are convenient, a more useful addition is the cross-reference utility. It lists functions and variables assigned and referenced within a program and has the capability to process **#include** files and multiple modules. To use the cross-reference utility, a special compilation option must be used to generate a cross-reference file.

The compiler operates in two distinct lexical environments: High C and draft ANSI. High C is a superset of Kernighan and Ritchie standard C; it includes extensions such as Pascal-like nested functions, underscores in constants (for readability), and the ability to mix declarations and statements anywhere in a function. The net effect of the High C extensions is to give High C a Pascal flavor. Because most of the extensions are in the areas where the Kernighan and Ritchie standard is vague, they cause little problem when writing "normal" C programs.

In addition to the High C version of the language, High C will compile any program that conforms to the proposed ANSI standard. This is accomplished using a compilation switch. Other switches to the High C compiler select memory model, object file name, cross reference, and list file generation. High C developers definitely need an aftermarket MAKE facility because it supports neither wild cards nor multiple file names on the command line.

The compiler supports five memory models for the PC. Both emulator and 8087 versions of each model's library are available. Programmers select models via either a compilation switch or a **#pragma** statement. Code can be generated for the 8087/287 math coprocessors or 80186/286/386 processors with **#pragma** statements. To use 386 protected mode, the object files must be linked with the Phar Lap linker and runtime environments.

High C supports the UNIX standard library and extensions such as **alloc** and **setjmp**. One unusual feature of this compiler is its "interface package," which handles primitive I/O functions—**read** and **open**—and not with library functions as they are in most C compilers. This way, users can write their own, albeit nonportable, implementation of these functions for embedded applications (those that do not use DOS, such as programs placed in ROM). In practice, using these interface routines in the benchmarks required a bit more rewriting than the other compilers needed.

High C has a flexible system for interfacing to other languages and libraries. Its **#pragma** statement can specify what sort of interface a routine has: call by reference, call by value, and called routine pops argument. This should allow High C to interface to most aftermarket C libraries or other programming languages.

If High C has a weak spot, it is in PC-specific library functions. It supports only the most rudimentary level of a access (interrupt 0x21 and basic file I/O). The user must write anything else or buy an aftermarket library.

The thick loose-leaf High C manual contains far more pages than it can actually hold gracefully. The documentation is organized into an Installation Guide, a Programmer's Reference section telling about the actual operation of the compiler and related tools, a Library reference, and a Language Reference describing all the extensions that have been made to High C.

The material seems complete, but technical in the best (or worst) UNIX tradition. Its strong areas are the installation guide and language description, which use syntax diagrams. It is weak in the library and interfacing sections. In addition, the tone is overly somber and the typesetting dense and confusing. For the high-level technical user who will use this product, however, the documentation is adequate. The manual

10 Important Reasons C Programmers Use Our File Manager

1. It's written in C.
Clearly the growing language of choice for applications that are fast, portable and efficient. All of db_VISTA's source code is written in C.

2. It's fast – almost 3 times faster than a leading competitor.
Fast access that comes from the unique combination of the B-tree indexing method and the "network" or direct "set" relationships between records. A winning combination for fast performance.

3. It's flexible.
Because of db_VISTA's combination of access methods, you can program to your application needs with ultimate design flexibility. Use db_VISTA as an ISAM file manager or to design database applications. You decide how to optimize run-time performance. No other tool gives you this flexibility without sacrificing performance.
db_VISTA is also well behaved to work with most any other C libraries!

4. It's portable.
db_VISTA operates on most popular computers and operating systems like UNIX, MS-DOS and VMS. You can write applications for micros, minis, or even mainframes.

5. Complete Source Code available.
We make our entire C Source Code available so you can optimize performance or port to new environments yourself.

6. It uses space efficiently.
db_VISTA lets you precisely define relationships to minimize redundant data. It is non-RAM resident; only those functions necessary for operation become part of the run-time program.

7. Royalty free run-time.
Whether you're developing applications for yourself or for thousands, you pay for db_VISTA or db_QUERY only once. If you currently pay royalties to someone else for your hard work, isn't it time you switched to royalty-free db_VISTA?

db_VISTA™

Features
- **Multi-user** support allows flexibility to run on local area networks
- **File structure** is based on the B-tree indexing method
- **Transaction processing** assures multi-user consistency
- **File locking** support provides read and write locks
- **SQL-based db_QUERY** is linkable
- **File transfer** utilities included for ASCII, dBASE optional
- **Royalty-free** run-time distribution
- **Source Code** available
- **Data Definition Language** for specifying the content and organization of your files
- **Interactive database access** utility
- **Database consistency check** utility

File Management Record and File Sizes
- Maximum record length limited only by accessible RAM
- Maximum records per file is 16,777,215
- Maximum file size limited only by available disk storage
- Maximum of 256 index and data files
- Key length maximum 246 bytes
- No limit on number of key fields per record
- No limit on maximum number of fields per record

Operating System & Compiler Support
- **Operating systems:** MS-DOS, UNIX, XENIX, ULTRIX, Microport, VMS, Macintosh
- **C compilers:** Lattice, Microsoft, IBM, Aztec, Turbo C, XENIX, UNIX and LightspeedC

8. db_QUERY & db_REVISE.
Add the SQL-based, ad hoc query and report writer for a relational view of db_VISTA databases.
Use db_REVISE to re-design your database easily and quickly!
Both royalty free!

9. Free tech support.
60 days of free technical and application development support for every Raima product. Of course, extended support and training classes are also available at your place or ours.

10. Upward database compatibility
Start out with file management in a single-user PC environment—then move up to a multi-user LAN or a VAX database application with millions of records. You'll still be using db_VISTA. That's why so many C programmers are choosing db_VISTA.

John, You forgot one...
11. db Vista training class
All you need to know to get the most from dbVista!
2-5 days...Basics, Advanced, and Internals.
Call Now! Will cut development costs a lot!
April 11-15

30-day Money Back Guarantee!
Try db_VISTA in your environment for 30 days and prove it to yourself. If not completely satisfied, return it for a

Price Schedule

	db_VISTA	db_QUERY
☐ Single user	$ 195	$ 195
☐ Single user w/Source	$ 495	$ 495
☐ Multi-user	$ 495	$ 495
☐ Multi-user w/Source	$ 990	$ 990
NEW:		
☐ VAX Multi-user	$ 990	$ 990
☐ VAX Multi-user w/Source	$1980	$1980

Order Now.
Put db_VISTA to work in your application program. Ordering is easy—simply call toll-free. We'll answer your technical questions and get you started. Call today.

Call Toll-Free Today!
1 (800) db-RAIMA
(800/327-2462) or
206/828-4636

MasterCard VISA

RAIMA™ CORPORATION

3055 - 112th NE, Bellevue, WA 98004 USA
(206) 828-4636 Telex: 6503018237 MCI UW

PT28

does not have a conventional index, but instead provides a traditionally UNIX-style, permuted key-word-in-context (KWIC) index.

The product posts slower than average compilation times, but the compiler performs optimization on the program before generating code. The execution times and module sizes are acceptable but not among the best for the compilers tested, so it would appear that the optimization is not terribly effective. (An article on code optimization is planned for an upcoming issue of *PC Tech Journal.*)

High C is a powerful, full-featured compiler. However, it compiles slowly, and the code it generates is not particularly small or fast. It will be most attractive for large projects that require the degree of extra control over a program that this compiler delivers.

Microsoft Corporation. In the last roundup of C compilers, Microsoft C came out as a solid product. It has only improved since then. Contained on nine diskettes with three thick loose-leaf manuals, it is a miracle of modern packaging, with envelopes, advertising flyers, and pamphlets falling out each time you open a manual.

The diskettes contain the compiler, model libraries, Microsoft's CodeView debugger, utilities, and linker. The utilities include LIB, a librarian; a MAKE utility; EXEPACK, a program to reduce the size of a .EXE file; and EXEMOD, a program to examine and change information in an .EXE file's header.

Along with the traditional command-driven Microsoft C compiler, the company has released its long-awaited QuickC compiler. Like Borland's Turbo C, QuickC is an integrated editor, compiler, debugger, and linker, and touts facility in using all of these tools and a fast, in-memory compilation.

A SETUP program automates installation and helps greatly for standard installations, but complicates custom installation. The process is slow, involving not only copying files but also building the common object libraries using Microsoft's librarian. The compiler grabs a lot of disk storage for all its models, making a hard-disk system, preferably an AT or better, a necessity.

The compiler supports most major compilation options and memory models. While it tends to compile and link programs slowly, it produces excellent object code. One minor omission from the selection of memory models is the tiny (or .COM file) model, which allows programs to contain 64KB of code and data in a single segment.

The documentation features an exhaustive user's manual, library reference manual, and a volume that describes CodeView and other utilities. Although voluminous, it is well indexed. Most of it has been broken up into smaller pieces for easy reading and searching. The assembly language interface, a trouble spot in other compiler manuals, is explained well here.

The version 5.0 compiler has added comprehensive optimization of the object code. New optimizations include in-line code generation, automatic assigning of registers to variables, loop invariant code removal, and better handling of expressions.

Our benchmark test results show that there are many contenders, but no single product was able to pull away from the pack.

The compiler's library provides only basic PC-specific support (interrupt and DOS call functions). In version 5.0, a graphics library has been added, but serious PC product developers will want an aftermarket library with more support (windows, graphics, and so on). With Microsoft C's widespread acceptance, the availability of add-on libraries is excellent.

Microsoft C does not come with a text editor, although the WordStar-like editor in QuickC could be used. Given the near-religious feeling most programmers have about their editors, this is not a serious omission. Microsoft provides a MAKE utility, but it is simple given the large-project orientation of this compiler, allowing minimal macro substitution and dependency listings. A large programming team might want to buy a more capable MAKE.

CodeView's debugging capabilities bolster Microsoft's already impressive product (For a review of CodeView, see "Multilevel Debugger, Mark S. Ackerman, March 1987, p. 90). It allows programmers to step through their source code, set breakpoints, and examine registers and stack frames. To use the debugger, programs are compiled with a special compilation switch that includes symbol and source line information in the .OBJ file. After linking the program to create a .EXE file, CodeView can be used to debug. The debugger is window-oriented, with a code window and a message window, and commands are activated via pull-down menus. The debugger, while not easy to use for a beginner, is a powerful adjunct to the compiler.

Paying $450 for Microsoft C buys a comprehensive, professional, quality compiler able to handle a job of any size. Only the relatively stiff price (for hobbyists and single users) and the slow batch-oriented operation of the optimizing compiler could be adverse factors in not choosing Microsoft.

HIGH MARKS

While good features and documentation can make a C compiler more flexible and easier to use, it is the object code size and speed that interests most hard-core C programmers. Our benchmark results show many contenders, but no single product was able to pull away from the pack.

The benchmark system used for this review was a 6-MHz PC/AT with a 30MB, 30-ms hard disk, 640KB RAM, and running DOS 3.2. The CONFIG.SYS file was present with FILES = 20 and BUFFERS = 20. An attempt was made to place data files in the same place on the disk to minimize variation in directory searching, and the system was rebooted between benchmarks to eliminate the effect of disk caching.

The benchmark results, summarized in table 4, include information from the following tests:

Compilation time benchmarks. These included compiling 60-, 150-, and 500-line benchmark files. The 500-line file pulls in three separate **#include** files. Turbo C, DeSmet C, Optimum-C, and QuickC had the fast compilation times for all the files tested; High C and C86Plus were the slowest.

Link time benchmarks. These tests involve linking one object file, then six. The compilers that did not provide linkers were timed using DOS 3.2 LINK; others, such as DeSmet DC88 and Aztec C, used their own proprietary linkers. Datalight and DeSmet did well here; Datalight's performance was most impressive because it was the only one of the three to use LINK.

Program sizes. The SIEVE and PENTATH programs measure size. SIEVE contains only integer library routines, while PENTATH incorporates floating-point routines. Both programs were produced for every compiler with the small memory model. Turbo C did well, producing the smallest .EXE file for SIEVE.EXE; so did Aztec, which pro-

duced the smallest PENTATH.EXE and the second smallest SIEVE.EXE.

Execution time benchmarks. Execution times were obtained by compiling and executing the AUTOx.C programs for both small and large models. The individual benchmarks in this suite include the Fibonacci test, which measures the basic overhead of function calls. The best performer here was Turbo C, although all the compilers did well.

The next two tests measure the speed of integer and long-integer math. Turbo C did the fastest integer math, and Lattice did the best with long integers (although not by much). Again, all the compilers did well in these tests, with only minor variation in times, even between small and large models.

The subscript test counts characters in a 500-character string, and the string copy test uses pointers to characters to copy one string to another. These are basic operations used repeatedly in C programs, so efficient code generation is important. The Eratosthenes Sieve can help determine the compiler's subscripting, integer math, and incrementing speeds. Both the pointer string copy and the Sieve were run with and without register variables. Some of the compilers, such as Lattice, did poorly with register variables in the large model, indicating that care must be taken when optimizing: the only real authority is your stopwatch.

The file I/O benchmarks copy a 30,000 byte file onto hard disks and diskettes using **getc/putc** functions, and the low-level **read** and **write** functions. In every case, the **getc** and **putc** calls were slower than **read** and **write**, which use larger buffers and make better use of DOS I/O facilities.

This is another test in which the times were highly variable, even between the same compiler's small and large memory models. Let's C and Datalight turned in the best **read/write** times, and respectable **getc/putc** times, although the writing is on the wall: use **getc** and **putc** in C programs for convenience, not for performance.

The floating-point tests include addition and multiplication (the dot product), **exp** and **log** functions, and **sin** and **tan**. Microsoft C 5.0, Turbo C, and Aztec C did well, with Aztec claiming the fastest add/multiply time and Turbo C having far and away the fastest transcendental and logarithmic functions, which compensates for the fact that its output .EXE files using the math library were bigger.

Numerical accuracy benchmarks. Introduced in the January 1988 article that was cited previously, these benchmarks cover multiplication, division, addition, subtraction, trig, and log functions of known quantities for known results to 17 digits. An aggregate error rating is reported, with lower numbers indicating a better performance: 0.0 to 0.5 is excellent, 0.5 to 1.0 is good, 1.0 to 1.5 is fair, and anything above that is considered poor. An increase of 1.0 in the result is an order of magnitude *worse* in numerical accuracy.

IN EXCELLENT HEALTH

Nearly every compiler excelled in at least one category, but in a price/performance comparison, Turbo C is hard to beat (turn to page 72 for a full review of Turbo C and QuickC). Microsoft C 5.0 was a fine performer overall in the execution time benchmarks, but it had rather sluggish compile and link times. It's easy to recommend Microsoft C for function, acceptance, and overall code performance, but many of the other C compilers reviewed here also have much to offer.

C Ware's DeSmet DC88 is a good choice for individual users who want a full-fledged development system at a low price. It includes handy tools, it compiles programs quickly, and it features accessible tech support. Mark Williams Let's C is another candidate for users in this category, especially beginners who may someday have to work with a command-line compiler and linker. Its shell, source debugger, and integrated editor provide a good learning facility without sacrificing compatibility in a more traditional edit/compile/link environment. And its documentation and support are first-rate.

Computer Innovations C86Plus performed well in the benchmarks, but had too many quality-control problems.

Manx Aztec C86 and Datalight Optimum-C are pleasant surprises, producing good code with a minimum of fuss. Aztec C86 is especially impressive because of its portability to many machines. Datalight's selectable optimizer and low price make it worth considering by individual users who need to "fine tune" single programs over a period of time.

Developers in C certainly have a luxury of choice.

Marty Franz is a programmer for Allen Test Products, a division of The Allen Group Inc., located in Kalamazoo, Michigan.

Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066-9987
408/438-8400
Turbo C
CIRCLE 342 ON READER SERVICE CARD

C Ware Corporation
945 Spring Street, Suite 14
Paso Robles, CA 93446
805/239-4620
DeSmet DC88 Development Package
CIRCLE 343 ON READER SERVICE CARD

Computer Innovations Inc.
980 Shrewsbury Avenue
Tinton Falls, NJ 07724
800/922-0169; 201/542-5920
C86Plus
CIRCLE 344 ON READER SERVICE CARD

Datalight Inc.
17505 68th Avenue NE, Suite 304
Bothell, WA 98011
800/221-6630; 206/486-8086
Optimum-C
CIRCLE 345 ON READER SERVICE CARD

Ecosoft Inc.
6413 North College Avenue
Indianapolis, IN 46220
800/952-0472; 317/255-6476
Eco-C88
CIRCLE 346 ON READER SERVICE CARD

Lattice Inc.
2500 S. Highland Avenue, Suite 300
Lombard, IL 60148
800/533-3577; 312/916-1600
MS-DOS C Compiler
CIRCLE 347 ON READER SERVICE CARD

Manx Software Systems Inc.
1 Industrial Way
Eatontown, NJ 07701
800/221-0440
Aztec C86
CIRCLE 348 ON READER SERVICE CARD

Mark Williams Company
1430 West Wrightwood Avenue
Chicago, IL 60614
800/692-1700; 312/472-6659
Let's C
CIRCLE 349 ON READER SERVICE CARD

MetaWare Inc.
903 Pacific Avenue, Suite 201
Santa Cruz, CA 95060-4429
408/429-6382
High C
CIRCLE 350 ON READER SERVICE CARD

Microsoft Corporation
P.O. Box 97017
Redmond, WA 98073-9717
800/426-9400; 206/882-8080
Microsoft C 5.0; QuickC
CIRCLE 351 ON READER SERVICE CARD

PC BRAND™ SPEAKS YOUR LANGUAGE

CLIPPER *From Nantucket*
Compile dBASE for Speed, Protection

Clipper™ turns lumbering dBASE® into a speed dMON with benefits bobbing in its wake: your source code is submerged from public view, you can distribute your compiled application without royalties, and your customers don't even need copies of dBASE! The Spring '87 Clipper offers index files compatible with dBase III Plus, and networking capabilities to run compiled programs on major networks supporting DOS 3.1 with no restrictions on number of users. Clipper offers arrays, menu-building commands, user-defined functions, context-sensitive help techniques for applications, a debugger, and it supports Expanded Memory. It goes well beyond dBASE with 1,024 fields per data base and 2,048 active memory variables.

Clipper has the power to save and restore multiple screens to and from memory variables. You can also create overlays, call object modules compiled in other languages, and create function libraries to link with your applications. Power and flexibility make it the #1 dBASE compiler. List: **$695**, PC Brand: **$375**.

McMAX *From Nantucket*
Like dBASE for the Macintosh

McMax™ is like running dBASE on the Macintosh. It combines an easy-to-use menu-driven ASSIST mode using the Mac interface, an interactive command mode like dBASE at the dot prompt, and an application programming language fully compatible with dBase III. It gives you the power to create dBASE language applications on the Macintosh and transfer back and forth to the IBM® world. McMax accommodates up to 16 million records, 32,000 characters per record, 255 characters per field, and up to 32 files open concurrently. No copy protection. List: **$295**, PC Brand: **Call**.

dBASE AT THE SPEED OF C
dBx Translates dBASE Applications to C

You dBASE™ programmers know what an expressive and readable language dBASE is. It's a very comfortable development environment. But the price is debased performance. Even compiled dBASE doesn't offer the speed that some users require these days. The kind of speed offered by software written in the C language. The answer is dBx™.

dBx translates dBASE to C. It offers you a major competitive advantage over the next dBASE programmer: Keep writing in dBASE. Take every application all the way to completion. Then use dBx to translate them top to bottom to C!

Other advantages: C is portable, even to other operating systems like UNIX/Xenix™. To the Macintosh or Amiga. dBx gives your applications a passport to places dBASE cannot go.

Has its own file manager for single user, but links to major C file managers—c-tree and dBC—for compatibility with dBASE files or multi-user support. We have everything you'll need, including good advice.

	List:	PC Brand:
dBx	**$ 350**	**$ 299**
with Library Source	**$ 550**	**$ 469**
with Full Source Code	**$1500**	**$1282**

dBC *Identical dBASE III Plus Files Using C*

dBC™ is a series of C libraries from Lattice which creates, accesses and updates files identical to those of dBASE itself. So dBASE can read and update the files too.

What for? It means both C and dBASE applications can operate on the same data bases interchangably. It means C programmers can interface with the big market of dBASE users out there, yet side-step the dBASE language. It means dBASE applications can now be linked to the universe of C libraries and tools to add windows, graphics, statistical analysis, all the things dBASE cannot do. It means the speed and power of C to impress clients accustomed to dBASE!

dBC's functions parallel all dBASE's file handling commands, many decomposed to permit direct data manipulation. Our versions of dBC mimic file formats for dBASE II and III and now dBASE III Plus makes your programs network ready!... as many stations as a network allows. Hands-off mode handles record and file locking and unlocking automatically. Close in functions give you direct lock/unlock control.

Supports all four memory models. dBASE II, III...List: **$250**, Ours: **$195**. dBASE III Plus...List: **$750**, Ours: **$595**. *Pay double and you get source too!*

THE SPINDRIFT LIBRARY
Fully Functional Fortran Library.

Spindrift's smooth package offers something previously unavailable to the Fortran buffs...a basket of functions, packed to the brim. No more tedious coding. Just call on Spindrift with it's armoury of functions.

Any number of Windows may be defined, each with a striking border and brilliant color. Define "pop-up" screens, Save/Restore images, set Cursor shape. Snare an entire screen into a Character array.

The Keyboard; read it without echo, or sense any keypress during execution. Cursor controlled directly with Edit keys.

DOS interface includes: Call System, Call Exec, Findfrst/Findnext for (*) and (?) file searching. Search the Path, subdirectory and file manipulation, command line argument parsing, date/time functions, DOS environment access, and much more. Other utilities also included.

Specify compiler, Microsoft or Ryan McFarland. List: **$149.00**. Us: **$129.00**

30-DAY MONEY-BACK GUARANTEE

We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out many programs themselves.** Ask your salesperson for details. There's just nothing stopping your buying from PC Brand.

C-WORTHY INTERFACE LIBRARY *NEW VERSION!*

The C-Worthy™ Interface Library wraps an entire user interface around your application. Its full power can be summoned by only a few high level calls. Sound exaggerated? A single function call can set up a complete text editor in a screen window. Recently acquired by Solution System, over 600 pages of Documentation, Turbo and Quick C version and a complete Interface Library have been added.

- High level calls pop menus and scrollable choice lists to the screen, restoring the background when dismissed.
- Windowing facilities open portholes of up to screen size for viewing virtual screens larger than the physical screen.
- Full context-sensitive help screen management takes over these chores and error messages. Automatic routines interrupt with pageable text windows explaining what to do next.

Novell found it "played a key role and accelerated development" in making its NetWare™ utilities easier for users. Ingenious demo: call for it.

Ask for:	List:	PC Brand:
C-Worthy	**$195**	**Call**
with Forms Library	**$295**	**Call**

BASTOC *BASIC Into C*

For a trifling price, BASTOC™ moves truckloads of BASIC code over to C. It's a translator which takes in Microsoft Extended BASIC and emits pure K&R C for Microsoft or Lattice. Structures even convoluted BASIC code. Optimized to dramatically reduce execution time. Dynamic string allocation ends BASIC's catatonic halts for garbage collection. Huge worksaver. List: **$495**, Us: **$399**.

BTRIEVE *B-tree File Manager Plus Add Ons*

If networks are on your horizon, betting your future on Btrieve as the one file manager for your C, Pascal, BASIC, and COBOL projects looks like a smart move. Reason? Novell bought Btrieve's creator.

Btrieve's function library takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, space recapture, forward and backward searching. Finds any key in a million in four or less accesses. Files may have up to 24 indexes; fixed record length to 4090 characters; variable length to 64k; indexes up to 255 characters; files of 4 billion bytes.

List: **$245**, Us: **$195**. Network Version List: **$595**, Ours Only: **$465**. Ask about Xtrieve for Btrieve file inquiry, Rtrieve for report writing. For any network that supports the MS-DOS 3.1 file sharing function.

POLYTRON VERSION CONTROL
Source Code Control for Any Language

PVCS allows programmers, project managers, librarians and system administrators to control the proliferation of revisions and versions of source code in software systems. Independent programmers, the leading software publishers and LAN companies, and hundreds of Fortune 1000 companies rely on PVCS to store and retrieve multiple revisions of text. It maintains a complete history of revisions as an "audit trail", generates status reports, and uses intelligent "difference detection" to minimize disk space for each new version.

On Corporate and Network PVCS simultaneous changes to a module are merged into a single new version. If changes conflict, the user is notified. The "Logfiles" used to track changes are interchangeable between any PVCS product.

Corporate PVCS is for multiple programmers. It includes "branching" to maintain code when programs evolve on multiple paths. Personal PVCS offers most of the power and flexibility of corporate PVCS, but excludes multiple programmer features. Network PVCS is the Corporate version for LANs. File locking and security levels can be tailored to each project.

PolyMake Still using a prehistoric Make? Step up to PolyMake, the most popular and powerful Make utility. Most flexible macros of any Make. Now integrated with PVCS and PolyLibrarian for faster performance and precise configuration management.

Ask for:	List	PC Brand:
Personal PVCS	**$149**	**$109**
Corporate PVCS	**$395**	**$309**
Network PVCS	**Call**	**Call**
PolyMake	**$149**	**$109**

PANEL PLUS
Library Source Code Gives It Complete Portability

There are no end of tools for screen design and data entry, but none quite like Panel Plus. Design a screen under program control, use Panel's utility to "run" and test it field by field, then pass it to Panel's code generator which delivers C source code. Options style the code to your compiler's liking, and you can of course do what you like to the source afterward. The code calls Panel Plus's function library, but now *the library comes in source*, so everything produced is highly portable. Not like other screen managers delivered as object libraries and which leave *you* to write the detailed code.

Panel Plus will operate in graphics mode via interfaces to graphics products it supports and can utilize the EGA's 43-line screen. Low-level I/O functions adapt it to various keyboards, screens, operating systems.

Panel's newest incarnation has every imaginable feature. A single screen design can have 1000 fields stacked as visual overlays up to 127 levels deep or as pop-ups. Groups of fields can be moved between levels. Screens can be output as compilable code or stored on disk for loading at run-time. Each field can be boxed, colored, multi-row, word-wrapped, and scrolled horizontally and vertically if larger than its on-screen view aperture. It can be assigned its own help and error message, can be told to accept certain characters, or to match a picture, and to check data after entry—proper dates, number ranges, etc.—using Panel's or your own validation routines. You can add your routines to Panel's test utility because even it comes as source. Fields are accessed in any order and control reverts to your application program after each field for choice of action.

For past Panelists, the new version has smaller and faster field and screen functions, tighter granularity, and an enhanced, reworked library. Major tool for the serious developer. List: **$495**, PC Brand: **$395**.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders or Literature, Call Us at....

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

3TJ11/87

PROGRAMMING PRODUCTIVITY TOOLS

BRIEF *Is Anything But.*
A Whopper of an Editor

With a name that belies its thoroughness, Brief™ has every feature you've ever contemplated for your editor-in-chief. Text in multiple buffers is scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text blocks may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers, with as many "undo" levels as you want.

Brief has text search abilities rivaling "grep", with wildcards for matching and indifference to intervening characters.

If you use Lattice, C86™, or Wizard, and have 320k, you can compile your C program without ever leaving Brief. It finds the lines with errors, and marches you through the text for repairs.

Parts of Brief were written with its own easy-reading Lisp-like macro language which has structure, conditionals, loops. "Simply the best text editor you can buy", Dvorak *Infoworld*. (Needs 192k.) List: **$195**, PC Brand: **Call.**

C-TREE & R-TREE
B-Tree File Manager Now Has Report Generator

c-tree: The only major b-tree file manager with network support in the standard low-cost version, c-tree™ gives you record-locking routines for DOS 3.1/3.2, UNIX and XENIX, and it even comes in C source code, yet there are no royalties. Source sticks to K&R, so C-tree is portable. Tests in many environments prove it.

Permits any number of keys for a data file—alpha, numeric, even floating point. Handles files with varied record lengths, multiple keys in one index file. Both high level and decomposed functions. It's the works.

r-tree: Adds the ability to produce ad hoc reports from files maintained by c-tree (v. 4.1 and up). Link a file description to the r-tree™ library, and use any text editor to write report scripts with no further C coding. Reports can access data in several files, select on criteria, join findings into new logical records, sort them, calculate new fields and columns, tabulate by control breaks. Comes in source, same portability as c-tree, and fits any compiler.

	List:	Ours:	Combined:
c-tree:	**$395**	**$329**	**$541**
r-tree:	**$295**	**$245**	

WINDOWS for DATA

FIRST PRIZE!

M'Soft Windows Compatible

"Only one package can be easily recommended" said *Computer Language* (June '87) reviewing nine window and data entry products for C. Complete field level functions specify prompt string, field length, data type, screen location, picture, target variable, entry rules, help messages, even functions to call for validation once data keyed in.

Windows for C is a subset. No data entry but all windowing functions. Unlimited windows can be made either to pop up or permanently overwrite the screen, scroll and highlight lists vertically and horizontally. Specify Compiler. Windows for Data: List **$295**, Ours **$259**. Windows for C: List **$195**, Here **$149**.

ESSENTIAL C UTILITY LIBRARY
400 Functions, 30c Each

You've probably seen the speed and power of Essential's C function library without knowing it. Software greats have been using it for some time to give today's top products pizazz and panache.

Now grown to 400 functions Essential produces pop-up menus, saves and restores screens and windows to disk or memory in as little as 1/10th second, and claims the fastest video output available. Library has 50 business graphics functions, 40 string handlers, 28 functions for printers, 18 for mice, 11 for time and date, DOS interface functions for disk error trapping, directory and file creation and management, lots more. Everything in source, including sample programs. We have versions with pre-built libraries for the well-known C compilers, and a source code librarian is supplied for rolling your own.

	List:	PC Brand:
C Utility Library	**$185**	**$119**
Essential Graphics	**$250**	**$183**
Essential Communications	**$185**	**$125**
with Breakout Debugger	**$250**	**$189**

Shopping List for the Power Workbench

	LIST	US
ARITY PRODUCTS		
Arity Combination Package	1095	979
Expert System Development Pkg	295	229
PROLOG Advanced Toolbox	150	119
PROLOG Compiler & Interpreter	650	569
Arity File or Screen Design Toolkits	50	44
SQL Development Package	295	229
Arity PROLOG Interpreter	295	229
Arity Standard Prolog	95	77
AI-EXPERT SYSTEMS		
EXSYS *Development Software by EXSYS*	395	309
EXSYS Runtime System	600	469
Insight 2+ *by Level Five Research*	485	379
AI-LISP LANGUAGE		
IQCLISP *by Integral Quality*	300	Call
IQLISP *by Integral Quality*	270	Call
Microsoft LISP *Common LISP*	250	199
TransLISP *from Solution Systems*	95	Call
TransLISP PLUS *from Solution Systems*	195	Call
AI-PROLOG LANGUAGE		
APT *Active Prolog Tutor from Solution Sys.*	65	Call
Turbo PROLOG *by Borland Int'l*	100	63
Turbo PROLOG Toolbox *by Borland Int'l*	100	64
ASSEMBLERS & DEBUGGERS		
Advanced Trace-86 *Morgan Debugger*	175	119
C-Sprite Debugger *by Lattice, source level*	175	139
Microsoft Macro Assembler *with Utilities*	150	109
PASM86 *by Phoenix, Macro Assembler*	195	138
Periscope I Debugger*...The Periscope Co.*	345	289
Periscope II *with NMI Breakout Switch*	175	139
Periscope II-X *software only*	145	105
Periscope III *with Advanced Board...New*	Call	Call
Periscope III 8 MHz *with Advanced Board*	995	795
Periscope III 10 MHz *Super-fast Version*	1095	875
BASIC LANGUAGE		
Microsoft BASIC Interpreter *for XENIX*	350	295
Microsoft QuickBASIC *...4.0*	99	63
BORLAND PRODUCTS		
Reflex & Reflex Workshop	200	129
Reflex *Data Base System*	150	89
Reflex Workshop	70	45
Turbo Basic*...New*	100	64
Turbo C Compiler*...New*	100	64
Turbo Lightning	100	64
Turbo Pascal & Tutor*...New*	125	85
Turbo Pascal with 8087 & BCD	100	64
Turbo PROLOG Compiler	100	63
Turbo PROLOG Toolbox	100	64
Turbo Tutor	40	28
C COMPILERS		
C86 PLUS *by Computer Innovations...New*	497	397
Lattice C Compiler *from Lattice*	500	299
Let's C Compiler*...w CSD Debugger*	75	54
MWC-86: Mark Williams C Development	495	369
Microsoft C Compiler *...4.0*	450	295
Turbo C Compiler *by Borland...New*	100	64
C INTERPRETERS		
C-Terp *by Gimpel Software*	300	249
Instant C *by Rational Systems*	500	395
Interactive-C *by IMPACC with debugging*	249	219
RUN/C Professional *from Lifeboat*	250	185
COMMUNICATIONS		
Asynch Manager *by Blaise, for C or Pascal*	175	135
Essential Communications *(See Essential Section)*		
Greenleaf Communications *by Greenleaf*	185	139
C UTILITY LIBRARIES		
Blaise C TOOLS PLUS/5.0	129	99
Blaise Turbo C TOOLS	129	99
Blaise C TOOLS PLUS	175	135

	LIST	US
C Food Smorgasbord *by Lattice*	150	109
C Utility Library *by Essential, 300 functions*	185	119
Greenleaf Functions *by Greenleaf Software*	185	139
COBOL LANGUAGE		
RM/COBOL*...see Ryan-McFarland Prod.*		
Microsoft COBOL Compiler	700	499
Microsoft COBOL Compiler *for XENIX*	995	795
Micro Focus COBOL*...see Micro Focus Prod.*		
dBASE SUPPORT		
BRIEF/dBRIEF*...Brief for DBASE III*	275	Call
CLIPPER*...from Nantucket*	695	375
McMAX*...dBASE for MAC from Nantucket*	295	219
Foxbase+ *...from Fox Software*	395	Call
Foxbase+ Multi-User	595	Call
APPLICATIONS PLUS*...Fox + Geller*	499	299
QUICKCODE PLUS*...Fox + Geller*	295	170
QUICKREPORT*...Fox + Geller*	295	170
QUICKENTRY*...Fox + Geller*	99	59
The UI Programmer*...Wallsoft*	295	244
The Documentor*...Wallsoft*	295	244
dFlow*...Wallsoft*	149	124
dBx*..dBASE to C Translator by DESKTOP AI*	350	299
with Library Source Code	550	469
with Full Source Code	1500	1282
dBC*...from Lattice...maintains DBASE files*	250	195
with source	500	390
dBC III Plus*...supports multi-user DBASE*	750	595
with source	1500	1185
DEVELOPMENT TOOLS		
BASTOC *by JMI, convert BASIC to C*	495	399
BASIC-C *BASIC's functions added to C*	175	139
Report Option *Softcraft Btrieve Report Gen.*	145	128
Xtrieve *Softcraft Query Utility for Btrieve*	245	220
Code Sifter *Profiler by David Smith*	119	89
Dan Bricklin's Demo Program *Prototyper*	75	69
LMK *from Lattice, "make" like UNIX*	195	149
Microsoft Window Development Toolkit	500	365
ESSENTIAL PRODUCTS		
C Utility Library	185	119
Essential Graphics	250	183
Essential Comm Library	185	125
with Breakout Debugger	250	189
FILE MANAGERS		
Btrieve *by Softcraft, no royalties*	250	195
Btrieve Network *by Softcraft*	595	465
C-Tree *by FairCom - no royalties, source*	395	329
R-Tree *by FairCom-Report Generator*	295	245
C-Tree & R-Tree Combo *by FairCom*	650	541
Opt-Tech Sort *Can sort Btrieve files*	149	105
XQL *SQL for Btrieve applications*	795	725
FORTRAN LANGUAGE		
RM/FORTRAN *by Ryan McFarland*	595	499
The Automated Programmer*...by KGK*	995	949
FORTRAN Libraries*...by Alpha Computer*	70	45
Microsoft FORTRAN *Links w/Microsoft C*	450	281
Microsoft FORTRAN *for XENIX*	695	546
Scientific Subroutine Package *by Alpha*	295	239
Spindrift Library *by Spindrift Laboratories*	Call	Call
GRAPHICS		
Essential Graphics *Essential, no royalties*	250	183
GSS Graphics Development Toolkit	495	375
GSS Kernel System *by Graphic Software*	495	375
GSS Metafile Interpreter	295	235
Halo *by Media Cybernetics*	300	219
with Dr. Halo II	440	299
Halo for Microsoft *includes all fonts*	595	434
MICRO FOCUS PRODUCTS		
Micro Focus Level II COBOL *w/Animator*	495	445
Level II COBOL	349	314
Level II Animator	195	175

	LIST	US
Micro Focus Personal COBOL	149	134
MODULA-2 LANGUAGE		
MODULA-2 Compiler Pack*by LOGITECH*	99	79
MODULA-2 Development Pack	249	199
MODULA-2 ROM Pkg & Cross RT Debugger	299	239
MODULA-2 Toolkit	169	139
MODULA-2 Window Pkg	49	39
OTHER LANGUAGES & UTILITIES		
Source Print*...source code formatter*	97	75
Tree Diagrammer*..source code diagrammer*	77	67
Microsoft Pascal Compiler *Links w/M'soft C.*	300	199
Microsoft Pascal Compiler *for XENIX*	695	546
Turbo Power Tools+ *by Blaise*	100	67
Advantage C++ *by Lifeboat Assoc.*	495	435
PHOENIX PRODUCTS		
C/PAC*...Pforce & Pre-C*	495	299
PASM86 *Macro Assembler*	195	98
PDisk *Phoenix's new disk manager*	145	89
Pfantasy Pac *six products...New Price*	995	549
PFinish *EXE performance analyzer*	395	194
Pfix86 Plus *Symbolic Debugger*	395	194
PforCe *vast library*	395	194
PforCe++*...Function Library for C++*	395	194
Plink86 Plus *Utilizes memory for overlays*	495	259
Pmaker *like UNIX "make"*	125	69
Pmate *with Macros*	195	98
Pre-C *UNIX "lint"-alike*	295	144
PTel *Binary File Communicator*	49	39
POLYTRON PRODUCTS		
PVCS Corporate*...Source Code Control Syst.*	395	309
PVCS Personal	149	109
PVCS Network	Call	Call
PolyMake *Complete Make Utility*	149	109
PolyLibrarian *Library Manager*	99	73
PolyLibrarian II *Library Manaer*	149	109
PolyShell *UNIX-like Command Shell*	149	109
PolyXREF *Complete Cross Ref Utility*	219	169
PolyXREF *One language only*	129	99
PolyBoost *The Software Accelerator*	80	64
PolyDesk III *3rd Generation Desktop Org*	99	73
PolyDesk III *Add On Tools*	Call	Call
RYAN-McFARLAND PRODUCTS		
RM/FORTRAN*...2.4 New*	595	499
RM/FORTRAN*...UNIX or XENIX*	750	599
RM/COBOL*...DOS*	950	799
RM/COBOL*...UNIX or XENIX*	1250	999
RM/COBOL 85*...ANSI 85 COBOL*	1250	995
RM/SCREEN*...for COBOL 85*	395	335
RM/NET +5*...COBOL & COBOL 85 Network*	300	249
SCREEN DESIGN		
Curses *by Lattice, UNIX screen designer*	125	99
with Source	250	199
Greenleaf Data Windows*...New*	225	169
with source	395	297
Panel Plus *by Round Hill, no royalties..New*	495	395
View Manager for C *by Blaise*	275	199
Vitamin C *by Creative Programming*	225	198
VC Screen *by Creative Programming*	100	81
Windows for C *Vermont Creative Software*	195	149
Windows for Data *includes Windows for C*	295	259
ZView *Data Management Consultants*	295	259
TEXT EDITORS		
Brief *from Solution Systems*	195	Call
Condor Editor *by Condor*	130	118
Edix *by Emerging Tech...Multi-screen*	195	159
Epsilon *by Lugaru Software, like EMACS*	195	149
FirsTime *by Spruce Technology, C syntax*	295	229
Kedit *by Mansfield, similar to Xedit*	125	99
LSE, the Lattice Screen Editor *Multi Wind.*	125	100
Vedit *by Compuview*	$1509 9	
Vedit Plus *by Compuview*	185	129

PRICED TO SAVE, SHIPPED FAST!!!

HALO

Graphics Tools for Software Developers

When choosing your graphics tools for software development, performance is crucial. The best proof of HALO's superior performance is in the number of Independent Software Vendors (ISVs) that develop and distribute HALO-based applications. Over 190 ISVs selected HALO because: HALO out performs the rest; HALO is easier to program and shortens time-to-market; HALO is complete with over 190 functions; 17 programming languages; and HALO supports a wider range of devices—over 110.

Since 1981 HALO has been the industry standard library of graphic subroutines for the PC. HALO has an installed base of 60,000+ end users and hundreds of site licenses.

A single copy of HALO includes all device drivers, your choice of one compiler binding, LearnHALO (an interactive tutorial), free 800 # technical support, and the license to use HALO on one machine. Flexible, practical licensing terms are available for ISVs and "Sites."

	List:	PC Brand:
Halo	**$300**	**$219**
with Dr. Halo III	**$440**	**$299**
Halo for Microsoft	**$595**	**$434**

ARITY'S

New Generation of Software Development Tools

Arity's advanced software technology brings to PC'S state of the art applications which previously required mini computers or special processors. Arity products combine not only the fastest benchmarks but C-Language integration, virtual memory, SQL, a sophisticated DBMS, presentation/window style interfaces, and expert systems to give your applications a competitive edge.

ARITY PROLOG Version 5, COMPILER/INTERPRETER or INTERPRETER

Most powerful Clocksin-Millish superset Prolog available. Is now the fastest, with a naive reverse benchmark of 17320 logical inferences per second on an 8 megahertz Compaq 286. Includes virtual database of up to 2 gigabytes with b-trees, hashing, built in editor and symbolic debugger, compile arithmetic, online help, over 200 predicates for windows, menus, dialog boxes, edit field, and list management. Version 5 supports embedded C expressions, global variables and structures, C data types, and calls to Prolog from C or vice versa.
List: **$650**, Us: **$569**

ARITY PROLOG ADVANCED TOOLBOX

Four groups of routines for building advanced applications with Arity Prolog version 5: a screen and report designer; file interchange routines for Ascii, 1-2-3TM, SymphonyTM, and dBASETM; a complete working spreadsheet; and miscellaneous data management routines.
List: **$150**, Us: **$119**

ARITY/SQL

A complete implementation of the IBM standard query language for OS/2. Arity/SQL maintains tables, indexes, and views. Queries can be accepted in SQL syntax or translated to SQL from menu selections or from another query language. Using Arity/SQL, tabular data can be extracted from any SQL compatible data base, manipulated in spreadsheets or PC data bases, then exported back. Available as a C callable library (requires a C compiler) or Prolog callable routines (requires Arity Prolog).
List: **$295**, Us: **$229**

ARITY/EXPERT

A set of advanced tools for building applications which approximate human decision making. Supports inexact reasoning, knowledge representation by either frame-based taxonomies or rules, interfaces to C and Pascal, 1 gigabyte of virtual memory and up to 20000 rules. Requires Arity Prolog.
List: **$295**, Us: **$229**

ARITY COMBINATION PACK

Groups all of the products described above, in a complete integrated development workbench.
List: **$1095**, Us: **$979**

GREENLEAF LIBRARIES

Functions

C source, assembler source, and binary libraries of 225 functions for many compilers. Emphasizes tight functional groupings to minimize loading code which your application may never use. Manual's 250 pages help select functions, as do demos, bulletin board.

A sampling: *DOS:* extensions for file and directory manipulation. *Screen:* select mode, page, monochrome or color, palette; cursor shape, positioning; clearing and scrolling; pixel get and put. *Strings:* efficient operations to add, delete, sort string pointers for top speed. *Other:* graphics primitives, keyboard status, function key assignment, time/date, read registers, get memory size, peek and poke.
Specify Compiler. List: **$185**, Here: **$139**.

Communications

Communicate with other users or remote data bases by asynchronous communications built right into your C programs! Over 120 functions *and demo programs in both C and assembler source code* set up separate transmit and receive ring buffers for up to 16 simultaneous channels. Interrupt driven so you can halt an incoming record, display it, file it, let the user edit it, then continue. Goodbye separate communications software.

Supports up to 9600 baud, ASCII or binary, any parity or word length, 8250 UARTs, Xon/Xoff and Xmodem, WideTrack receive. Specify compiler.

List: **$185**, Us: **$139**.

GSS GRAPHICS SYSTEM

Leave the Device Driving to GSS

GSSTM has reconfigured two components of its comprehensive graphics tools to conform with the ANSI Computer Graphics Interface (CGI) standard.

At the heart of the system is the Development Toolkit which contains all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras, and more. Drivers house management of vector graphics (plotters) and bitmaps used by raster input devices (scanners) to insulate the application program from concern for device idiosyncracy. No one else has implemented CGI that way. It means your programming remains generic; just switch drivers and the same program will drive a different device.

GSS KernelTM conforms to level 2b of ANSI's Graphical Kernel System (GKS) and contains all its needed drivers and language bindings. Kernel has macro level tools to draw and color an object, store the sequential instructions, and recreate the object on its own, as well as segment it, transform it, etc. So powerful, a single command may represent several score lower level statements.

Kernal has the tools for graph and chart generation and their captioning: hand it apples and oranges, say "pie", and it bakes the numbers into a digestible display for screen or plotters.

Kernel can convert the images it creates to ANSI Computer Graphics Metafiles (CGMs), a tokenized standard for storing every form of graphic image as data. The Metafile Interpreter reads the contents of a CGM and interprets it with full CGI capability for recreation on various devices.

Quality software? IBM thinks so. They sell the GSS series under their own label.

Unit royalties and annual fees have been instituted for redistribution. Needs 256k.

Ask for:	List	PC Brand:
CGI Dvlpmt Toolkit	**$495**	**$375**
Kernel System	**$495**	**$375**
Kernal for IBM RT	**$795**	**$645**
Metafile Interpreter	**$295**	**$235**

BLAISE C TOOLS PLUS/5.0

C TOOLS PLUS/5.0 from Blaise Computing Inc. helps you to quickly build professional applications using the full power of Microsoft C 5.0 and QuickC. Now you can concentrate on program creativity by having full control over DOS, menus, interrupt service routines, memory resident programs, fast direct video access; windows; printer and keyboard control, and more!

Blaise Computing's attention to detail, like the use of full function prototyping, cleanly organized header files, and a comprehensive, fully-indexed manual, makes C TOOLS PLUS/5.0 the choice for experienced developers as well as newcomers to C.

C TOOLS PLUS/5.0 prebuilt libraries are ready to use with either QuickC or the Microsoft C 5.0 command line environment. Complete documented source code is included so that you can study and adapt it to your specific needs.

	List:	PC Brand:
C TOOLS PLUS/5.0	**$129**	**$ 99**
Turbo C TOOLS	**129**	**99**
C ASYNCH MANAGER	**175**	**135**
Turbo POWER TOOLS	**99**	**75**
Turbo ASYNCH PLUS	**99**	**75**

The AUTOMATED PROGRAMMER

Enter Mathematical Notation Directly

A programming system for numerical computation that enables a computer to recognize and process mathematical, engineering, and scientific specifications. Using the Two-dimensional (2D) Text Editor, input computational formulas directly using conventional mathematical notation (e.g., integral, summation, numerator-over-denominator division, matrix inverse, dot and vector products, etc.) and technical English. The description of the problem solution (program) is expressed at such a high level that it approaches the ideal of an executable specification, i.e., for many problems, *the specification is the program.* A correct program can be achieved many times faster than by current programming methods. Verification and debugging are often reduced to proof reading—"If it looks right, it is right."

The Automated Programmer translates 2D programs into a target language, permitting easy interfacing with existing program libraries. The currently available target language is FORTRAN.

Computational results may be embedded in complex 2D text and graphic "IMAGE" formats using intuitively obvious (WYSIWYG) control.
List: **$995**; PC Brand: **$949**

MICROSOFT C 4.0

The Keeper of the Technology Takes Over

It bundles a source debugger and a "make", and sports a "huge" memory model permitting single data objects larger than 64k, but what's really impressive about Microsoft C are the benchmarks reported in Dr. Dobb's. Microsoft runs away from a field of 17 winning 11 of 27 benchmarks.

The CodeViewTM debugger uses windows to show everything on one screen: source alongside disassembled object, variables, stack and registers. Drop down windows obviate learning of commands. "A source-level debugger that puts the rest to shame" said Dobb's.

Microsoft C has five memory models for code and data, plus non-library support for another thirteen, and boasts alternate math packages for speed versus accuracy, with or without 8087/80287 chips.

Both linker and library manager are part of the package, as is the "make", which knows how to rebuild any size project by compiling only elements which have changed.

It is reportedly used by Lotus, Ashton-Tate and, fittingly, Microsoft itself to develop Windows. Dobb's calls it "the best MS-DOS C development environment value today [for] virtually any kind of program conceivable." 320k suggested.

List: **$450**, PC Brand: **$295**

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.
Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders or Literature, Call Us at....

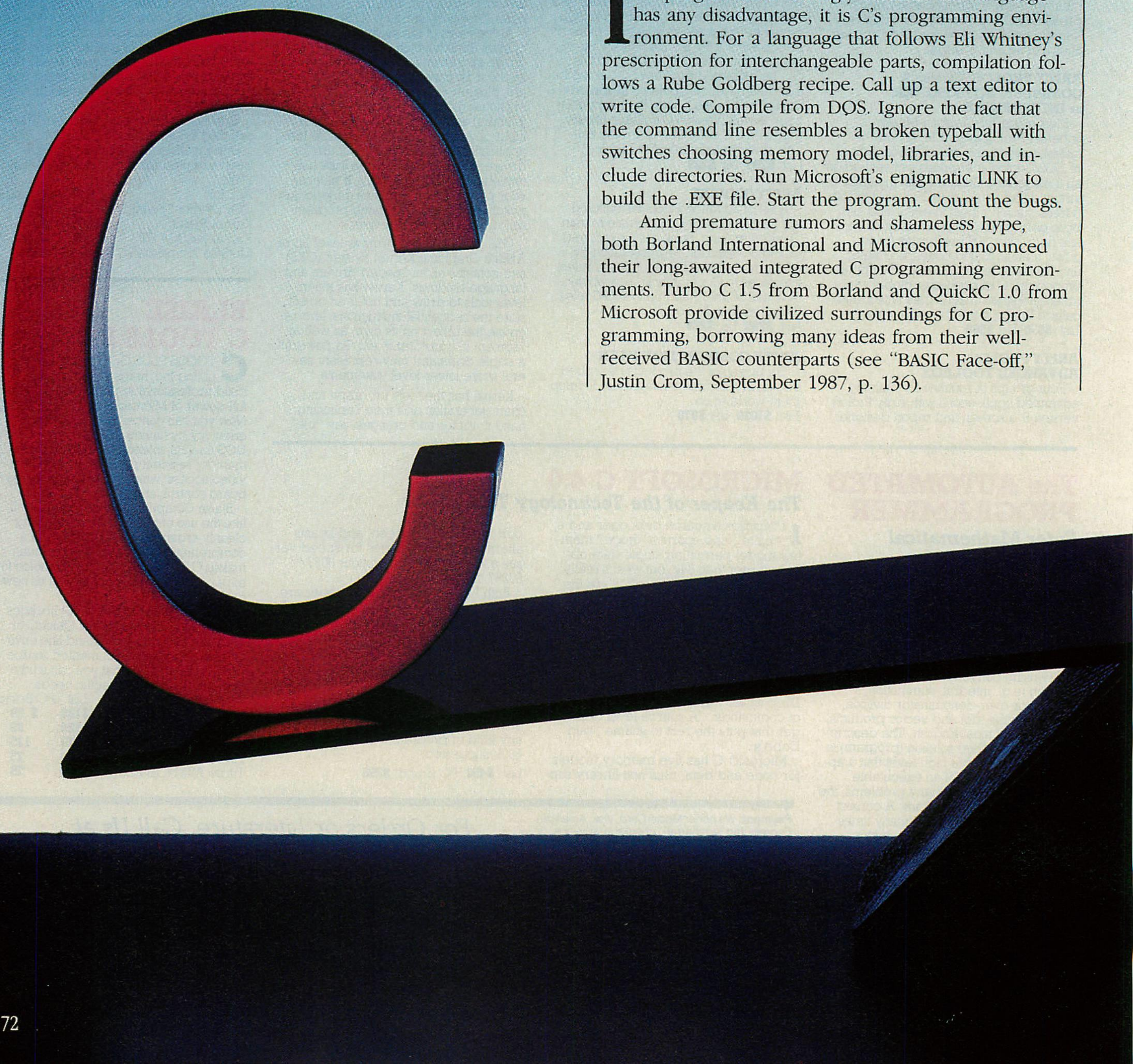
800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

Turbo and Quick Weigh in

If C programmers willingly concede the language has any disadvantage, it is C's programming environment. For a language that follows Eli Whitney's prescription for interchangeable parts, compilation follows a Rube Goldberg recipe. Call up a text editor to write code. Compile from DOS. Ignore the fact that the command line resembles a broken typeball with switches choosing memory model, libraries, and include directories. Run Microsoft's enigmatic LINK to build the .EXE file. Start the program. Count the bugs.

Amid premature rumors and shameless hype, both Borland International and Microsoft announced their long-awaited integrated C programming environments. Turbo C 1.5 from Borland and QuickC 1.0 from Microsoft provide civilized surroundings for C programming, borrowing many ideas from their well-received BASIC counterparts (see "BASIC Face-off," Justin Crom, September 1987, p. 136).



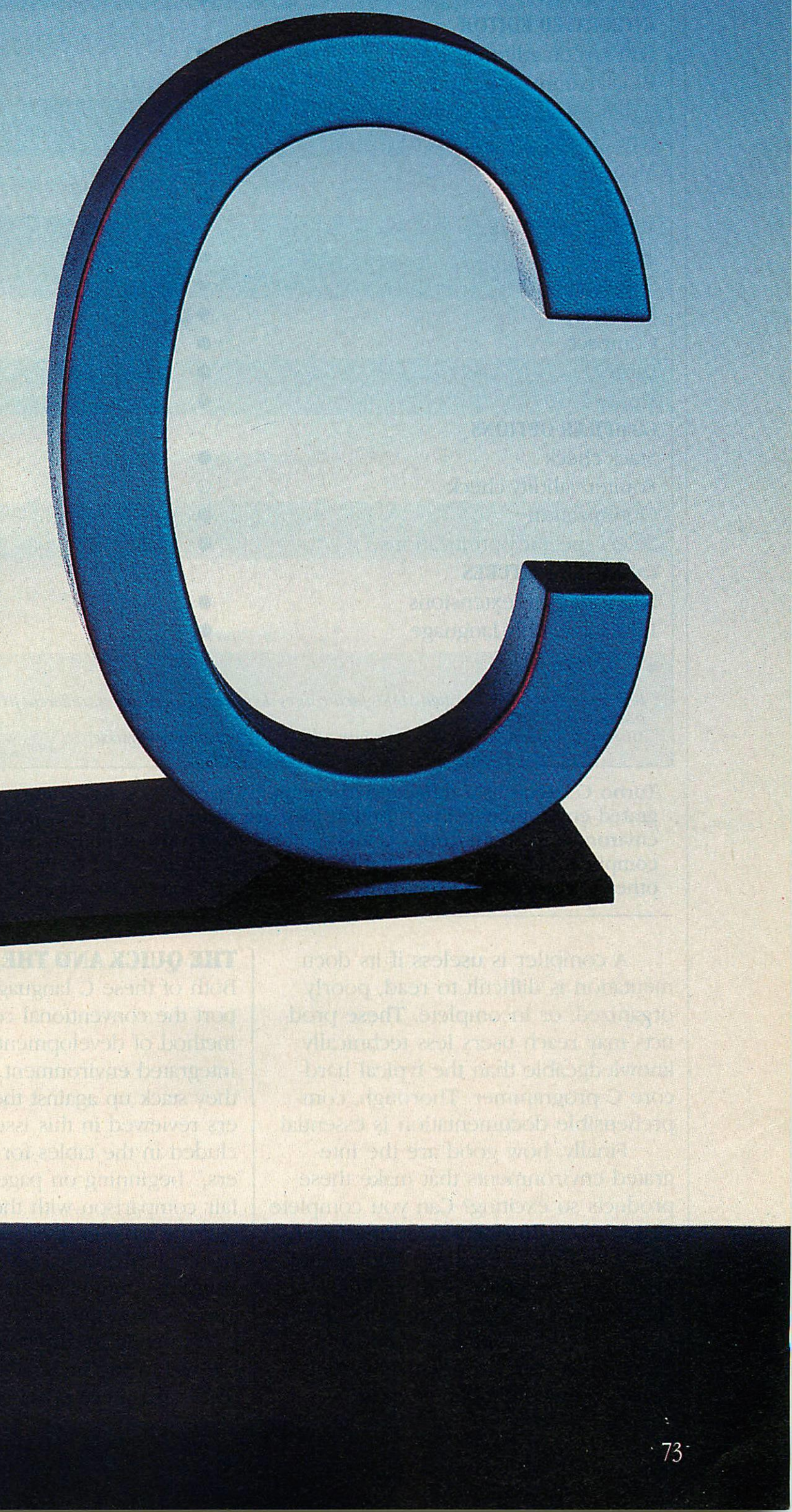
C programmers have a tough balancing act in trying to choose between the integrated environments of Turbo C and QuickC.

MARTY FRANZ

In judging the two compilers, a fair amount of attention needs to be paid to language support. Flashy environments often sacrifice linguistic power for a convenient compilation environment. Are these compilers good enough to write real programs? Do they support a complete and standard implementation of the C language? Does the compiler support the portable UNIX standard library and the draft ANSI library, yet provide PC-specific functions to make programming for the PC family easier?

The C language emphasizes portability and adherence to standards across different systems; a dazzling program environment that implements only a subset of C would be unacceptable to most C programmers.

The compiler should produce compact, speedy code. We will look at the performance of Turbo C and QuickC, putting them through their paces with the standard *PC Tech Journal* benchmark suite.



PHOTOGRAPHY • MARC DAVID COHEN

TABLE 1: *Compiler Specifications*

	BORLAND	MICROSOFT
PRODUCT	Turbo C	QuickC
VERSION	1.5	1.0
PRICE	$99	$99
FEATURES		
Minimum RAM required	384KB	448KB
Integrated MAKE facility	●	●
Stand-alone MAKE facility	●[a]	●
Integrated debugger	○	●
Linker	●	●
Librarian	●	●
INTEGRATED EDITOR		
Full-screen editor	●	●
Block commands	●	●
Find replace (case sensitive)	●	●
Find replace (whole word)	●	●
Mouse support	○	●
On-line help	●	●
MEMORY MODELS		
Tiny	●	○
Small	●	●[b]
Medium	●	●
Compact	●	●[b]
Large	●	●[b]
Huge	●	○
COMPILER OPTIONS		
Stack check	●	●
Pointer validity check	○	●
Optimization	●	●
Select specific optimizations	●	○
LANGUAGE FEATURES		
Supports MSC extensions	●	●
Inline assembly language	●	○

● = *Yes* ○ = *No*

[a] *Borland's Turbo C integrated MAKE facility uses dependency descriptions different from the stand-alone MAKE program.*
[b] *Microsoft's QuickC integrated environment supports only medium model.*

Turbo C excels as a full-featured compiler, but QuickC has a more capable integrated environment. Even though they are distinctive because of their integrated environments, both Turbo C and QuickC are worth examining as command-line compilers. Additional information about their features and comparisons with the other C compilers reviewed in "C Contenders" can be found in table 1 (p. 56).

A compiler is useless if its documentation is difficult to read, poorly organized, or incomplete. These products may reach users less technically knowledgeable than the typical hard-core C programmer. Thorough, comprehensible documentation is essential.

Finally, how good are the integrated environments that make these products so exciting? Can you complete large projects in them, working indefinitely, editing, compiling, linking, and debugging programs without entering a single line at a DOS prompt? Do they reduce the time needed to create a program as compared with the conventional compile-edit-debug cycle?

THE QUICK AND THE TURBO

Both of these C language products support the conventional command-line method of development as well as an integrated environment. To show how they stack up against the other compilers reviewed in this issue, they are included in the tables for "C Contenders," beginning on page 52. To obtain a fair comparison with the other compilers, the table entries and benchmark results in that article were obtained using the command-line versions of Turbo C and QuickC.

In contrast, this article focuses on the use of the integrated environments that make these products stand out from the pack of conventional C compilers. Functionally, the Turbo and Quick environments accomplish the same thing. They both provide an integrated editor, compiler, and linker featuring easy use of all these tools and fast, in-memory compilation (see table 1). They rate very closely in several aspects so they make for a difficult choice. They represent, however, two different styles, and different users may find one easier to use than the other.

Both compilers support the full UNIX and draft ANSI C language standards, plus Microsoft extensions such as **near**, **far**, and **pascal** key words; most programs should port between the two environments with little effort. The two compilers have different names for many PC-specific DOS and BIOS access functions, but both offer the same programming functionality.

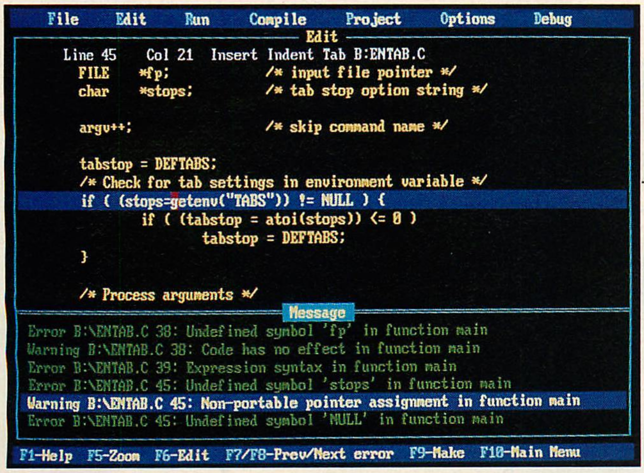
OUT AND ABOUT IN TURBO C

Borland ships Turbo C with six diskettes and three paperback manuals. The diskettes contain integrated and command line versions of the compiler, linker, librarian, a MAKE utility, five model libraries, a graphics library, include files, and sample programs. A UNIX-like **grep** utility is also provided.

Installation is simple: build the directories described and copy files from the diskettes. A batch file is provided to facilitate building the required directories. The primary configuration file, TURBOC.CFG, can be created to tailor the command-line version of the compiler to find directories for include files, libraries, and the compiler support files. A separate configuration file, TCCONFIG.TC, stores compiler settings for the integrated environment; it is changed through menu selections. The TCINST program can further customize many aspects of Turbo C's integrated environment, including window sizes, screen colors, editor key functions, and editor mode settings.

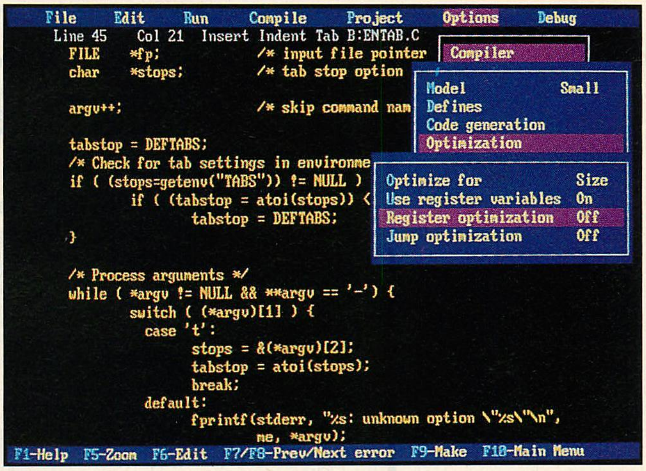
With the default settings, the layout of the Turbo C screen appears as shown in photo 1. A menu selection bar appears at the top of the screen. The remainder of the screen is divided into a large editing window and a smaller window that contains messages from the compilation and linking process. The two windows are coupled such that scrolling through the message window and selecting an error message automatically positions the edit window to edit the file at the line where the error was generated. This makes fixing errors in large, multifile programs a nearly effortless process.

PHOTO 1: *Turbo C Screen*



The Turbo C screen contains, from top to bottom, a menu selection bar, an edit window, and a message window.

PHOTO 2: *Turbo C Nested Menus*



Options for both the Turbo C compiler and the runtime environment can be set through a series of nested menus.

The six pull-down menus—File, Edit, Compile, Project, Options, and Debug—are selected by pressing F10, followed by the initial letter of menu title. A command for Run shares the menu bar. Along with the ability to select menu options with the cursor keys, shortcut keys are available, such as Alt-R for Run or Alt-D for Debug.

The File menu has commands for loading and saving files, DOS directory functions, and exiting to a shell. A useful function is Pick, which displays the most recently visited files and makes hopping between files easier.

The Edit menu drops you into the Turbo C editor. The integrated WordStar-like editor is similar to the one Borland has included in all its products. It is unable to edit two files at once, and has the restriction that the entire source file fit in memory. Otherwise, it is fast and smooth in operation. Macros would be a welcome addition, although a keyboard enhancer, such as Borland's SuperKey or RoseSoft's ProKey, can do the trick.

The Project menu sets or clears the current project file. Project files comprise a simple MAKE facility, and their contents are straightforward: one file each per line, with any dependent files (such as header files) next to it enclosed in parentheses. This organization, while convenient, does not offer the power of macro-based MAKE facilities. It also will not rebuild custom libraries with a librarian. It is adequate for most users and for small- to medium-sized programs. A traditional and more powerful MAKE program is included in Turbo C to support the command-line version of the compiler.

The Run option, available on the menu bar, activates the Make facility (if a project file has been defined) and will compile and run the current program. Another route to the MAKE facility is through the Compile menu. It optionally compiles the current file only, links the current file only, or rebuilds the entire program.

The Options menu (see photo 2) makes the Turbo C integrated environment really shine: all the options for the compiler and linker that are normally specified as hard-to-remember command-line switches are available in nested pull-down menus. Choosing the memory model, **#define** tokens, selecting the level of error reporting, and determining the program's calling sequences, all can be done without touching a single switch or **#pragma** statement. The options, once set, can be saved to the configuration file.

Finally, the Debug menu activates Turbo C's message-tracking facilities. This is not a real source-level debugger, as the label might indicate. Errors are logged to a message file. You can scroll through the file to select the specific error to examine, then press a key and jump to the point in the source file at that particular line. While of little help for runtime errors, this makes debugging compiler errors simple.

Behind the beauty of the Turbo C environment stands the brawn of a full-fledged compiler. In addition to supporting the draft ANSI language standard, Turbo C offers enormous flexibility with its selectable warning levels and optimizations. You can, for example, optimize for size or speed, ignore register declarations, disable warnings about pointer assignments, and in short do everything possible with more expensive compilers, including compiling programs with Pascal rather than C calling sequences.

Borland's Turbo C compiler will generate programs in all six memory models from the integrated environment as well as from the command line. Various pointer and function types also can be mixed if needed. A number of small improvements to Kernighan and Ritchie have also been made, as well as support for the draft ANSI Standard. The manual carefully documents these. One useful addition is inline assembly language code.

In addition to the normal UNIX and ANSI functions, the Turbo C library has a few extra functions for the PC environment, such as BIOS calls, interrupt handling, and directory and file handling. Recently, third-party libraries have appeared for Turbo C, so add-on availability is not an issue. Optional source code for the runtime library is available from Borland for $150.

Turbo C 1.5 introduced a well-endowed graphics library, possibly as a reaction to the graphics library of QuickC. Borland's initial ante in the screen management area is impressive; it includes text-based windowing functions as well as a graphics-based routines. It supports IBM adapters (EGA, CGA, MCGA, and VGA), Hercules monochrome adapters, and the 640-by-400 pixel graphics used by AT&T, Olivetti, Compaq, and Toshiba. Borland claims the routines, formally named the Borland Graphics Interface (BGI), will be supported in other Borland products such as Turbo Pascal 4.0.

Fourth Annual Novell Developers Conference Application

February 17, 18, 19   Salt Lake City, Utah

Please fill me in on the Fourth Annual Novell Developers Conference and the Novell Developers Program.

Name ______________________________

Title ______________________________

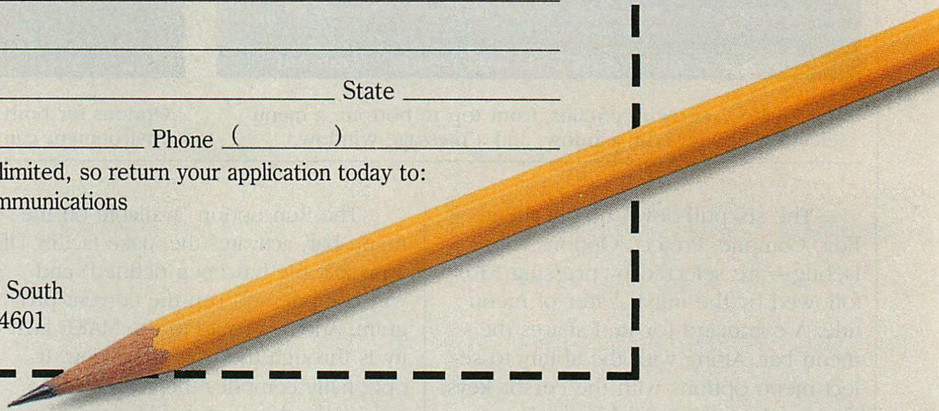
Company ______________________________

Address ______________________________

City ______________________ State ________

Zip ____________ Phone (   ) ____________

Attendance is limited, so return your application today to:
Corporate Communications
Novell, Inc.
Box 5900
122 East 1700 South
Provo, Utah 84601



Fill in this application, and we'll fill you in on what's new for Novell applications.

While you're developing applications to run on Novell networks, we're developing new directions for those networks. You could spend a lot of time just keeping yourself up-to-date. Or you could spend three days at the Fourth Annual Novell Developers Conference and get up-to-the-minute information on future Novell technology.

At the conference, you'll hear first-hand about Novell's plans for product advances and enhancements. We'll reveal the most promising directions for multiuser applications. And you'll have the chance to share ideas and solutions with other Novell developers. Seminars will address a wide range of topics, including:

- Universal NetWare® Architecture update
- Open Protocol Technology Architecture
- OPT implementations
- Value Added Processes
- System Fault Tolerant Technology update
- Network management and diagnostics
- OS/2
- NetWare 386
- SQL
- Message Handling Service (MHS)
- LU 6.2
- HLLAPI & PCOXAPI
- Wide Area Communication APIs
- TCP/IP

Send in your application today and learn what's coming from Novell tomorrow. But hurry, attendance is limited. Return your application to Corporate Communications, Novell, Inc., Box 5900, 122 East 1700 South, Provo, Utah 84601. Or for more information, contact Corporate Communications by phone (801) 379-7808 or FAX 801-375-0735.

NOVELL

For network solutions, you should be seeing red.

To speed-up linking, Borland includes a LINK-compatible linker, TLINK (Turbo Link), with Turbo C. The linker is compact (10KB) and fast, but it is not completely compatible with Microsoft LINK, instead it follows the Intel .OBJ format religiously and ignores comment records, which LINK uses.

There are a few nits to pick with this otherwise comfortable product. The compiler can be finicky about **breaks** and **returns** used in **switch** statements to the point at which warnings about these must be turned off lest a deluge of messages result.

Turbo C's first manual, the *User's Guide*, includes setup instructions, information on how to use the various parts of the package, and tutorial information. Its *Reference Guide* contains the traditional library, include file, and error message information.

The third Turbo C manual, *Version 1.5 Additions and Enhancements*, documents features such as the graphics library, the librarian, and new installation options. The inclusion of function upgrade information in the third manual requires that a user refer to two separate manuals when searching for function information; there is no master index for both manuals.

Do not think Turbo C's paperbacks mean Borland gives the language lean documentation. Quite the contrary. The 307-page *User's Guide* introduces the Turbo C compiler and environment, providing a brief but useful tutorial for Turbo Pascal programmers. It is fairly good, with an excellent section explaining memory models and variable storage. However, beginning programmers still will need a basic primer on C.

The organization of the 405-page *Reference Guide* resembles that of other C compilers, with lots of functions and header files. It includes tables of compiler and runtime messages and a summary of the operation of the environment. Each library function is well documented. It includes a section on portability to other compilers and systems, and even compatibility with Turbo Pascal. Most entries include examples. Turbo C also provides a competent on-line Help facility, but it is a refresher, not a substitute for the manuals. On the whole, however, the documentation is outstanding.

If Turbo C has a major weakness, it is in runtime debugging: its integrated environment has no built-in debugging facility, and there is no standalone debugger. The linker can generate .MAP files for debuggers such as Microsoft's CodeView or The Periscope Company's Periscope, but local variables (those on the stack and in registers) will not be identified by name.

Beyond the lack of integrated debugging, Turbo C is an impressive, full-featured compiler. At $99, it represents a legitimate value.

A QUICK STUDY

Microsoft ships QuickC as either a stand-alone product or as part of the C 5.0 compiler. When used with C 5.0, it can serve as a prototyping tool for the more comprehensive (but slower compiling) optimizing compiler. QuickC's mix of convenience and performance may leave some C 5.0 buyers wishing they had saved their money and bought only the stand-alone version of QuickC.

As a stand-alone product, the package consists of five diskettes, three paperback manuals, and a quick-reference card. A setup program performs the installation, which includes linking a set of libraries to match requested options, such as 8087 emulation or inclusion of the graphics library. Because the QuickC integrated environment works only with the medium memory model, that model should always be installed. Other models may be installed if the command-line compiler is being used. Environment variables are used to specify the location of include, library, and temporary files.

Instead of pop-up menus and control keys, QuickC displays dialogue boxes with buttons and controls, all selected via the mouse.

QuickC is not as adaptable as Turbo C for different personal preferences. A menu option can be used to change the colors of some screen components, but menu colors cannot be changed. Users of monochrome monitors with color display boards will find that highlighting on menu text is not visible unless QuickC is started with the /B (black-and-white) option. Horizontal and vertical scroll bars can be removed if not desired. Settings for both screen colors and compiler features are placed in a file called QC.INI, which is placed by default in the current directory. If the settings should be applied to QuickC wherever it is started from, the file can be placed in a directory along the PATH (usually wherever QC.EXE resides).

The QuickC integrated environment (see photo 3) resembles QuickBASIC's. A Microsoft-compatible mouse is not needed, but it makes using the environment easier. In contrast to Turbo C, which is optimized for fleet-fingered typists, QuickC is closer to a text version of Microsoft Windows or a Macintosh application; instead of pop-up menus and control keys, QuickC displays dialogue boxes with buttons and controls, selected via the mouse. Shortcut function keys are available.

The QuickC environment includes a File menu for selecting files and projects; an Edit menu that controls copying, cutting, and pasting of text blocks (QuickC also has an Undo facility); a View menu for selecting source and include files for editing; a Run menu for compiling and executing programs; a Search menu that includes some **grep**-like pattern-searching capabilities, and a Debug menu.

One noteworthy feature of QuickC is the Edit Program List function of the File menu, which makes building and maintaining project files especially easy. A pop-up window displays a list of files, and all you need to do is select the ones you want included in the project. This builds a basic .MAK (project) file, which can be customized later for more exacting tasks. The built-in MAKE facility includes powerful macro substitution commands.

Like Turbo C's editor, QuickC's editor bears a strong resemblance to WordStar and can edit only one file at a time. The View menu includes an option to open the last file viewed. This allows alternating between two files to clip text or check information.

QuickC avoids deeply nested menus by using large control panel displays to set options. A good example is the Run-Compile control panel, shown in photo 4. The display is quite lucid and easy to set using a mouse. The procedure for setting options with the keyboard are not as intuitive: the Tab key is used to move between fields; the space bar or arrow keys are used to set the option, depending on the control being set.

When a file is compiled in the QuickC environment and has errors, a message window is opened at the bottom of the screen to display error and warning messages. Menu or function-key commands can be used to scroll through the errors, positioning the cursor at the error location in the source

LASERVIEW PLUS: THE EDITORS' CHOICE IS NOW SUPPORTED BY ALL THE STANDARDS.

". . . unquestionably altered the definition of high resolution."
PC Week

". . . provides the sharpest image of all."
PC World

"LaserView: The Master of the Raster"
Bits & Bytes Review

". . Breathtaking . . . the Rolls Royce of large screen monitors."
InfoWorld

What They've Seen:
LaserView, the ultra-high resolution, 2,000,000 pixel (1664x1200) display system for IBM PC, XT, ATs.

The ultimate system for displaying two full facing pages of legible text and graphics for your publishing applications.

Or detailed design drawings without excess panning for CAD/CAM applications.

What You'll Get:
LaserView PLUS,™ everything the editors raved about, PLUS a whole lot more: IBM-Monochrome and Hercules compatibility. And now LaserView stands alone. Literally! You can run all your standard software without a second monitor. So the premium display for professional graphics applications is now the only display you'll need.

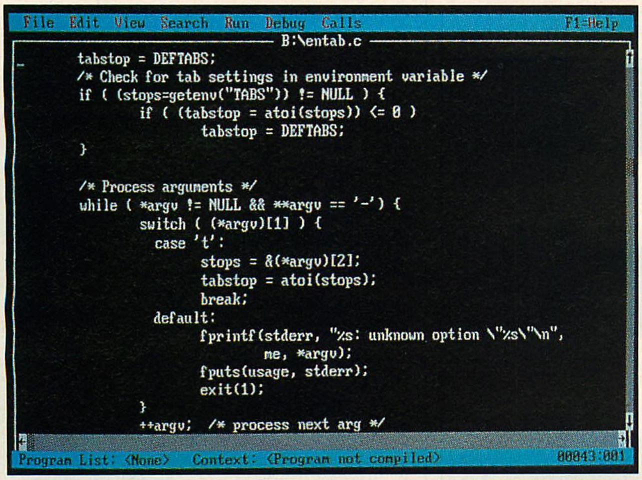
For details and the dealer nearest you, call 415-770-0100.

Sigma Designs, Inc.
46501 Landing Parkway
Fremont, CA 94538

S I G M A D E S I G N S

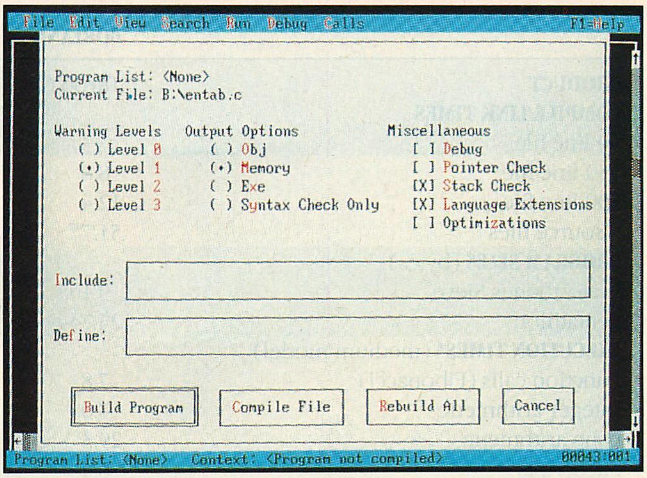
CIRCLE NO. 202 ON READER SERVICE CARD

PHOTO 3: *QuickC Screen*



QuickC's spartan screen contains only a menu bar, scroll bars, and a bottom line that displays the current status.

PHOTO 4: *QuickC Dialogue Box*



In contrast to the nested menus of Turbo C, QuickC compilation options are set through a single, large dialogue box.

file. Another command is used to close the error message window when it is no longer needed.

QuickC excels in debugging programs. The integration of the debugger into the environment is superb. Its debugger is a whittled-down version of Microsoft's CodeView debugger and shares the same interface. It can single-step and trace programs, setting watches on variables and expressions. During a trace, a moving highlight shows the line in the program being executed. The stack of called routines can be displayed. QuickC provides for smart pointers, a feature that compares pointers with hazardous memory addresses to prevent memory corruption. All of these features allow convenient, interactive debugging.

The linking process in the QuickC integrated environment works differently than in the command-line compiler. QuickC has 143 functions built into the environment, including the standard I/O (**stdio**) library, the string family (**str**) routines, and the memory (**mem**) routines. Any routine not built into QuickC must be explicitly loaded, either from the standard libraries or from a Quick Library (.QLB) file.

Quick Libraries are used when in-memory compilation is selected. They are created by writing a stub source file that calls every routine needed by the real source file (or files). The stub file is then compiled and linked with the standard library (.LIB) files to yield a .QLB file. To use the Quick Library, QuickC is started with an option to load the named .QLB file. When compilation to a .EXE file is selected, QuickC runs an external linker to search the standard library files and load separately compiled object files. Most users will prefer compilation to a .EXE file; it is more convenient, albeit slightly slower, than using Quick Libraries.

QuickC's library is extensive, containing graphics and other PC-specific programming tools. The library is common with C 5.0, so programs are easily moved between the two environments. The QuickC graphics library has support for the CGA, EGA, and VGA graphics modes. It includes routines for setting screen modes; drawing points, lines, objects, and shapes; and filling figures. This is a good basic library for beginning applications programmers but it relies on a simple object depiction about as sophisticated as interpreted BASIC. For most professional applications, a more complete aftermarket graphics library will be needed.

The 420-page *Programmer's Guide* provides information on using the command-line compiler and the integrated environment. QuickC's myriad features are a disadvantage here; the manual hops from a "quick start" overview of the environment, over to a too-brief introduction to C programming and the built-in graphics library, back to a more detailed view of the environment, then on to a description of the QuickC command-line utilities and assembly language interfacing. The manual is poorly organized for QuickC-only use and, like Turbo C's, is not suitable for novices.

QuickC's 246-page *Language Reference* is a concise, no-nonsense description of the C language. Users looking for a tutorial for C, however, should look elsewhere. An appendix details the differences between QuickC and the Kernighan and Ritchie standard.

The *Run-Time Library Reference* uses its first 100 pages to give an overview of the library calling conventions, categorize the major library routines, and describe the contents of library include files. The remaining 585 pages are an alphabetically ordered reference to the library routines. Although portability is mentioned briefly in the overview, individual descriptions of routines do not cover portability to other systems. Most entries include an example of correct usage.

In return for QuickC's convenient editing and debugging environment, programmers give up a few C 5.0 compilation options and advanced features, such as greater code optimization, the CodeView debugger, and huge memory model. For many programmers the omitted features are not crucial. The QuickC compiler is very compatible with C 5.0, even down to the MAKE (.MAK) files, which are extremely easy to use in QuickC's environment. Furthermore, QuickC's debugger is much more convenient than CodeView because it is built into the editor.

For debugging programs in other than the integrated environment's medium model, CodeView can be used with programs compiled and linked by conventional command-line utilities. Through an option switch, QuickC will generate the debugging records that allow CodeView to symbolically display local and register variables.

Because CodeView is not provided with QuickC, a separate Microsoft product, such as the Macro Assembler, would have to be purchased to obtain

TABLE 2: *Benchmark Results*

	BORLAND	MICROSOFT
PRODUCT	Turbo C	QuickC
COMPILE/LINK TIMES		
60-line file	7.3	8.0
150-line file	8.8	13.7
500-line file	12.4	13.4
8 source files	51.7[a]	49.3[b]
PROGRAM SIZES (bytes)		
Eratosthenes Sieve	9,108	10,777
Pentathlon	25,744	27,494
EXECUTION TIMES[c] (medium model)		
Function calls (Fibonacci)	7.8	7.9
Integer arithmetic	7.0	9.6
Long arithmetic	29.3	29.0
Subscripts	8.4	9.7
Pointer use	12.8	15.4
With register variables	36.7	54.8
Eratosthenes Sieve	5.1	8.7
With register variables	31.5	36.0
FILE I/O		
Read/write[d]		
Diskette to diskette	6.7	6.2
Hard disk to hard disk	0.8	0.8
Getc/Putc[e]		
Diskette to diskette	63.4	50.9
Hard disk to hard disk	11.8	11.6
FLOATING-POINT OPERATIONS		
(Software/80x87 chip)		
Add/multiply (dot product)[c]	24.5/3.1	23.9/3.2
Exponential/logarithmic[f]	7.4/1.0	14.0/1.3
Sin/tan (trigonometric functions)[f]	8.4/1.1	14.7/1.4
NUMERICAL ERROR RATING[g]	0.9	3.7[h]

All times in seconds. *[a] 3,255 lines of code. [b] 2,874 lines of code. [c] 20 iterations. [d] 1 iteration. [e] 2 iterations. [f] 10 iterations. [g] Smaller values indicate higher numeric accuracy. [h] See text.*

Both Turbo C and QuickC live up to their names for compilation and execution time. Timings were taken in each product's integrated environment using the medium memory model. Compile-and-link times were taken using each program's MAKE feature; both products were directed to compile to a stand-alone .EXE file.

the debugger. Even so, the purchase price of both products ($249) is half the price of C 5.0 alone, which does not include an assembler.

POTENT-C

The *PC Tech Journal* benchmark suite used in "C Contenders" evaluated the performance of the compilers. Combined compile plus link times were measured with a stopwatch from the time the keyboard command was issued from the integrated environments until control returned to the keyboard after linking. The medium memory model was used for benchmarking because this is the only model supported by the QuickC integrated environment. Programs were compiled under QuickC with stack overflow checking disabled, as this is the default for Turbo C. All other options for both compilers were at their default settings. Benchmark results are presented in table 2.

On the programs tested, Turbo C is faster than QuickC in compiling and linking, although QuickC is still faster than most of the other C compilers tested. Turbo C also enjoyed an advantage over QuickC in code size. Both the PENTATH and SIEVE benchmarks were smaller in Turbo C. In execution times, Turbo C either matches or beats QuickC on every benchmark, but its advantage is very slight in most cases. Turbo C's performance may be slightly understated in these tests because many Turbo C users will be able to take advantage of the more efficient small memory model.

The numeric accuracy benchmark revealed a bug in QuickC: constant computations were performed only to float accuracy, not to the required double. When notified of the problem, Microsoft promised to have it fixed in time for the next release of QuickC (due out in February). Because of the bug, the numeric accuracy rating was a dismal 3.7; however, by changing the benchmark to avoid the bug, QuickC obtained a respectable rating of 0.8.

Both QuickC and Turbo C are excellent products. However, Turbo C has the edge as a more powerful implementation. The one feature the Turbo C product covets is an integrated debugger, such as the one QuickC has. The built-in QuickC debugger closes the edit-compile-debug cycle; a program development environment cannot be truly integrated without the debugging component. QuickC's demerits come from the limited memory-model support and cumbersome operation for users without a mouse.

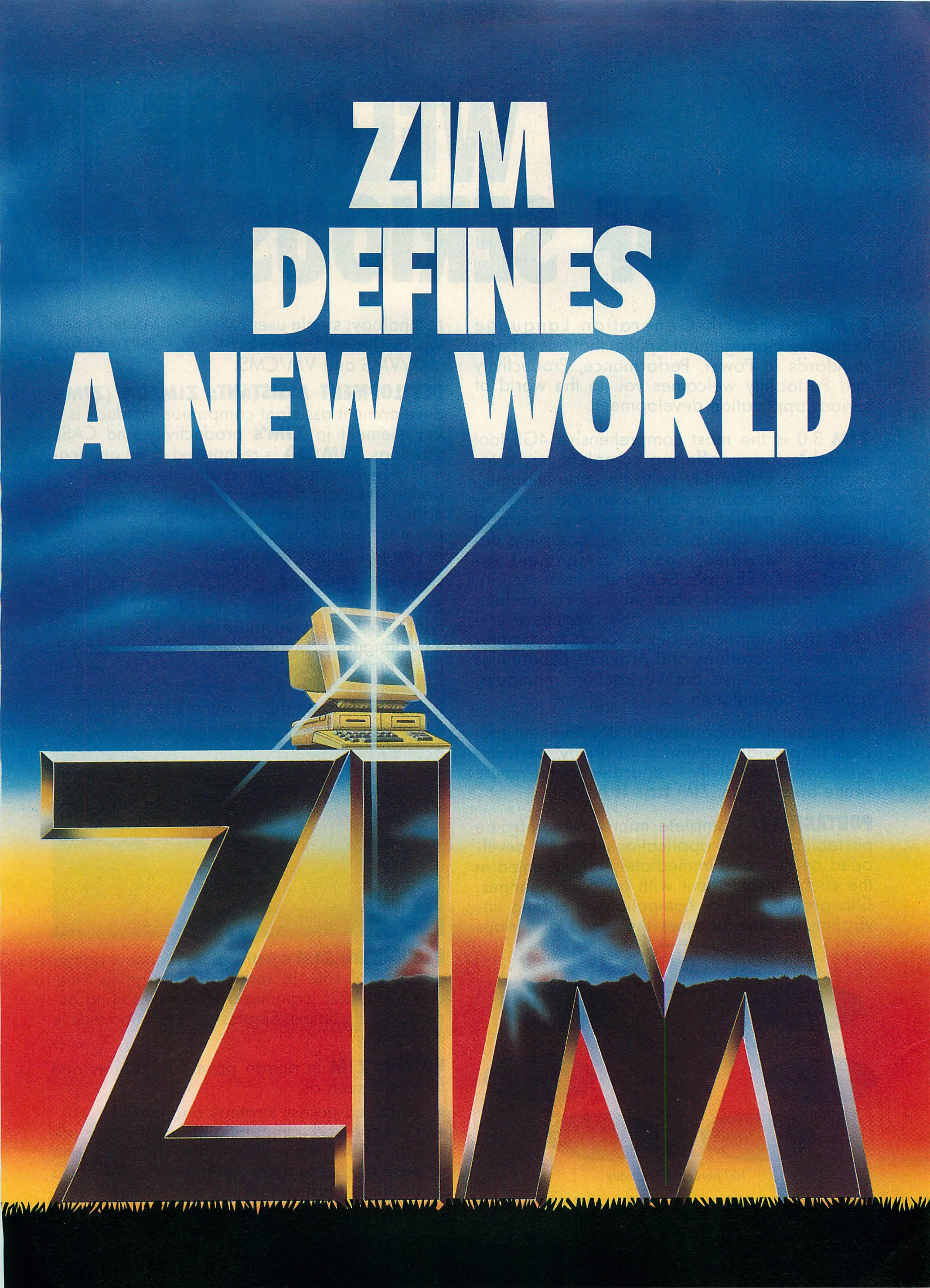
Do not overlook that both products provide conventional command-line compilers and linkers and perform well as such. The prices of these packages and their innovative environments make them both worth experimenting with, if only to decide that integrated programming environments are not quite ready yet.

Borland's Turbo C and Microsoft's QuickC both fail in making C approachable to the novice. Streamlining the C programming interface does not die the complexity of the language, and the beginning C programmer is still going to be baffled by the memory models, compilation switches and calling sequences. For experienced C programmers, however, choosing between one of these two excellent offerings is a delightful dilemma.

Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066
800/543-7543; 408/438-8400
Turbo C 1.5
CIRCLE 335 ON READER SERVICE CARD

Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073
800/426-9400; 206/882-8080
Microsoft QuickC 1.0
CIRCLE 334 ON READER SERVICE CARD

Marty Franz is a programmer for Allen Test Products, a division of The Allen Group, Inc., located in Kalamazoo, Michigan.


ZIM
DEFINES
A NEW WORLD
ZIM

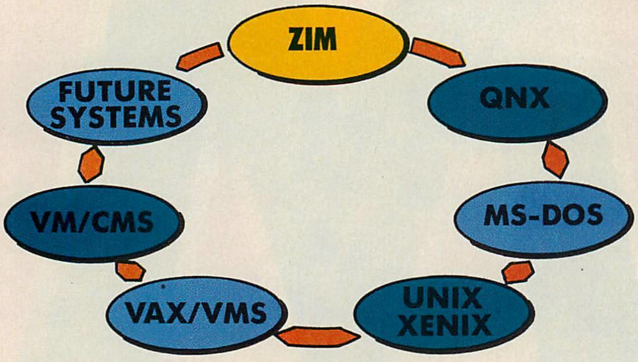
WELCOME TO
OF SERIOUS 4GL/

ZIM, the Fourth-Generation Language Database Management System that has set new standards in Power, Performance, Productivity and Portability welcomes you to the world of serious application development.

ZIM 3.0 is the most comprehensive 4GL tool available today, offering complete micro-to-mainframe portability (with no code changes), performance optimized in each operating environment, multi-user, extensive import/export capabilities, rapid prototyping, complete debugging facilities, nested assignment expressions, CASE tools, SQL, multi-level security, report painting and more. All in a comprehensive procedural language that permits everything from simple queries through to complex database operations and provides capabilities for comprehensive data extraction, manipulation and presentation.

To be useful, an application development tool **has** to provide maximum flexibility, performance and productivity—regardless of the nature of the application. ZIM **has it** NOW!

PORTABILITY: Complete micro-to-mainframe portability means applications can be developed on PC's and immediately implemented in the client environment with no code changes. Our growing list of compatible operating environments encompasses the world of serious technology: single user MS-DOS, MS-DOS networks, Novell networks, QNX, UNIX, XENIX, VAX/VMS and VM/CMS.



100 percent portability

DEVELOPMENT ASSISTANT: ZIM/DA (ZIM's development assistant companion product) is a key element in **ZIM's** productivity and CASE offerings. **ZIM/DA** is comprised of an application generator, menu generator, active and comprehensive data dictionary, report painter, code editors and context-sensitive help. **ZIM/DA** permits remarkably fast transition from prototype to finished application.

THE ZIM LANGUAGE: Time is money. It shouldn't be spent writing endless lines of code or worse, tying up expensive mainframe systems. The **ZIM** language, based on the entity-relationship (E-R) model, provides the power that serious application developers demand. Compare a typical SQL command and the **ZIM** equivalent:

SQL:
```
SELECT *
FROM WORKONTAB, PROJECTS, EMPLOYEES-
WHERE WORKONTAB.ENUM=EMPLOYEES.ENUM-
AND WORKONTAB.PNUM=PROJECTS.PNUM-
AND PROJNAME='ALPHA'
```

ZIM:
```
LIST ALL EMPLOYEES WORKON PROJECTS WHERE
PROJNAME='ALPHA'
```

DEBUGGING: ZIM's active data dictionary plays an integral role in reducing debugging time. Use the PARSE command to verify coding accuracy and context. Singlestep execution aids in debugging.

POWER: ZIM is rich in powerful features and functions such as:

- AI-based access strategy analyser.
- Unsurpassed computational abilities, including CASE expression which handles with ease the 'step' functions found in real-world business applications.

THE WORLD
DBMS PROGRAMMING

- Natural multi-user features including user-defined transactions, roll-back, audit trails and roll-forward.
- Complete code-data independence.
- Exceptional text-handling capabilities.
- Unlimited forms manipulation and report-writing capabilities.
- Multi-lingual applications (French, German etc.)

SECURITY: ZIM's multi-level security features include log-in, passwords, permission-based access to entity sets, relationships and fields and encryption/decryption of entity sets and relationships.

VIRTUAL FIELDS: ZIM's unique virtual fields capability allows you to easily implement composite or concatenated keys, derived or computed fields, soundex indexes, etc.

IMPORT/EXPORT: ZIM masters the importing and exporting of screens and data to and from word-processing, spreadsheets and provides superlative forms composition through laser printers. **ZIM** applications already in use have saved millions of dollars through the reduction and/or elimination of paper forms.

ZIM is supported by an ever-growing family of companion products like **Niva's Guide to ZIM** (the authoritative guide from introductory to expert levels), the **ZIM Compiler** to ensure the highest level of performance of your **ZIM** application, a **Program Language Interface** permits "C" access to **ZIM** databases, **ZIM-ISQL** allows queries based on SQL syntax, **Runtime** and **Query Runtime** systems and **ZIMPLE,** an add-on module that permits easy use, allows end-users to create their own simple applications, ad hoc queries or paint their own reports (used with **ZIM** or **Query Runtime**).

And you'll be supported too! Zanthe's Priority Support program includes first-rate telephone technical support, free updates, new versions and periodic technical bulletins. **Code Review** and **Training** services are available for all levels of **ZIM** users.

Around the world, from Australia to Europe, **ZIM Service Centers** and **Distributors** stand ready to supply and assist any **ZIM** user.

Already an integral part of data management in national governments, international banks, financial corporations and thousands of other applications **ZIM** welcomes you to the world of serious application development.

Zimpressive!

Memory in the Hot Seat

The performance burden now shifts to memory as it tries to keep pace with faster 386 processors. This requires improved memory architecture.

STEVEN ARMBRUST and TED FORGERON

Just as a chain is no better than its weakest link, so overall performance of a personal computer is limited by its slowest device. As time goes by, technology improves computer system microprocessors and devices, shifting the performance bottleneck from one area to another.

The new generation of 80386 machines has put memory performance in the hot seat, as systems designers struggle to find ways to get memory to keep up with the lightning-fast CPUs.

When IBM introduced the PC in 1981, diskette drives overshadowed all other elements of the PC in holding back performance. With the introduction of the PC/XT and its 10MB hard disk, disk performance improved to a point where system designers could concentrate on other areas, namely the XT's relatively slow 4.77-MHz 8088 and its narrow (8-bit) data path.

The PC/AT raised the speed to 6 MHz with its 80286, and provided a 16-bit path for data to and from the CPU. To match the improved speed and memory access of the 286, IBM provided 20MB and 30MB drives with 40-ms access times. This boost in disk performance kept pace with added CPU and memory performance.

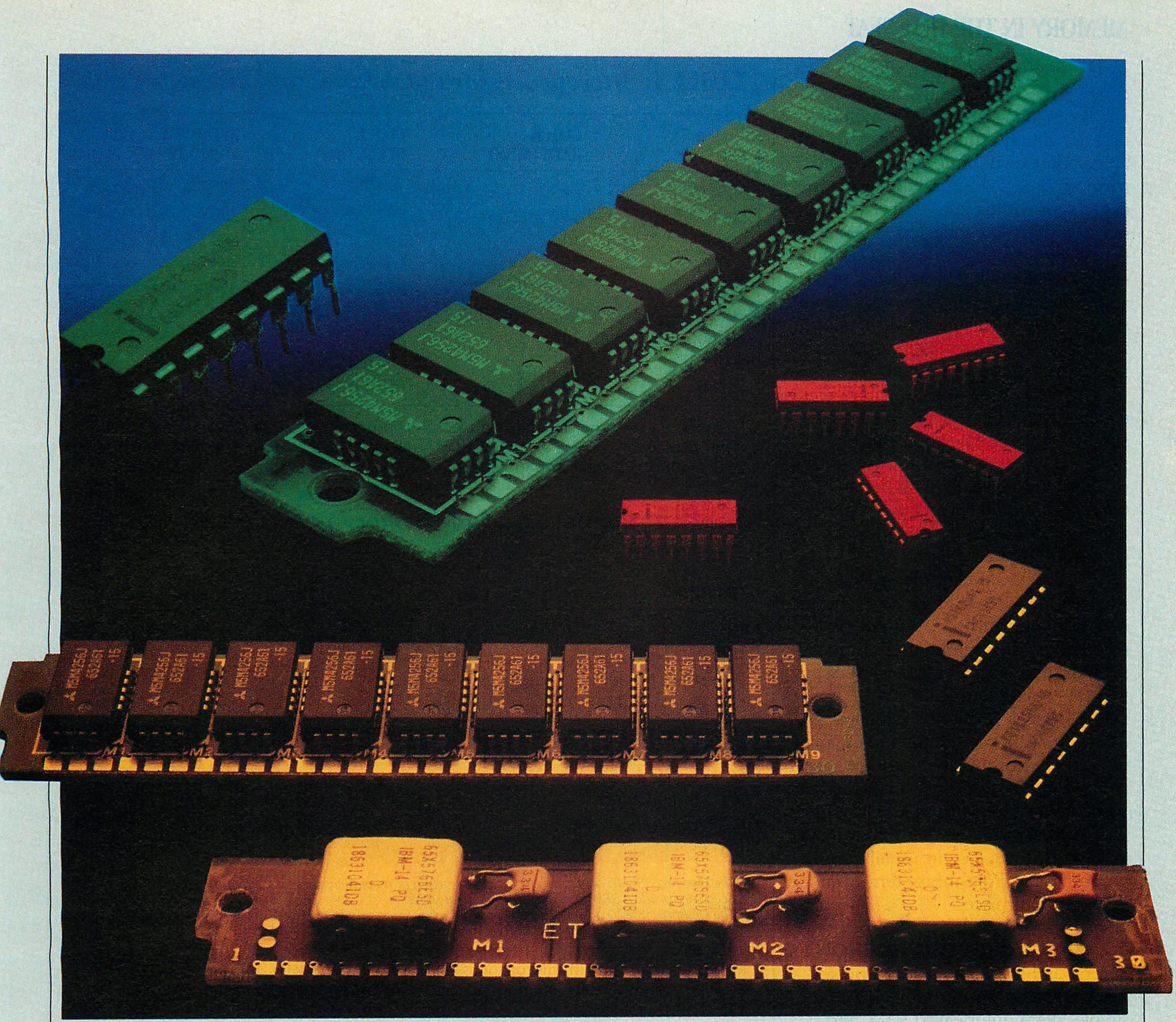
With the advent of 80386-based machines running at 16 MHz, 20 MHz, and beyond with a 32-bit data path to memory, the performance bottleneck has shifted from disk and CPU to memory itself. Even though memory speeds have increased, the processor is so fast that in a system with a directly connected dynamic RAM (DRAM) system, the processor spends an inordinate amount of time simply waiting for information it requests from memory.

Because approximately 70 percent of all bus cycles are code fetches, the processor often waits for code to execute. This is equivalent to a high-performance race car losing traction so often because of its slick tires that it does not move much faster than cars with smaller engines.

HOW FAST IS FAST ENOUGH?

Memory access certainly can affect overall performance. On Intel 286 and 386 processors, a memory access requires a minimum of two clock cycles, one cycle for memory to accept the address of the location to be accessed, and a second cycle for it to provide the contents of the location.

The processor is not necessarily inactive during these clock cycles, because it has separate bus and execution units. While the bus unit is retrieving information from memory, the execution unit can perform other operations, such as manipulating registers. If a con-



flict occurs, the execution unit has priority over the bus unit. However, the processor cannot access information from memory until the bus unit spends the necessary clock cycles obtaining it.

The time required for the processor's bus unit to obtain information from memory is the shortest possible memory access time. Both 8088 and 8086 processors require four clock cycles for a memory access. The 286 and 386 processors require two clock cycles. The two clock cycles take different amounts of time on different processors (even the 286 and 386) because different processors operate at different speeds. Therefore, a processor's optimum memory access time depends not only on the number of clock cycles required for a memory access, but also on the clock-cycle length.

For example, a clock cycle for a 4.77 MHz 8088 is 210 nanoseconds (ns) and the processor requires four clock cycles for a memory access, therefore, the 8088 requires 840 ns before it can accept the data it requested from memory. On the other hand, a clock cycle for a 20-MHz 386 is 50 ns. The 20-MHz 386 requires only two clock cycles for a memory access, so it can accept data 100 ns after requesting it.

Because the processor spends approximately 70 percent of its bus cycles fetching code, the ideal memory system is one in which the rate that memory can supply information to the processor matches the rate that the processor can execute code. If memory is slower than the processor, the system is said to be *bus bound*. If the processor is slower than memory, the system is *processor bound*. Making either the processor or memory faster than the other will not improve performance, but will probably cost more.

If memory cannot respond fast enough to meet the processor's demand for data, the processor must wait one or more clock cycles, literally doing nothing, until memory is able to respond to the microprocessor's request. Each clock cycle that the processor has to wait is called a *wait state*. Memory fast enough to respond to the CPU in two clock cycles is said to operate at zero wait states. Memory requiring three clock cycles operates at one wait state, and so on. Table 1 lists zero-wait-state access time and clock-cycle length for several CPU clock rates. Table 2 lists clock rates of several computers and the number of wait states required to access memory.

The goal of all system designers is to provide memory that is fast enough to match the execution speed of the processor. Zero-wait-state memory satisfies (and even exceeds) this memory

PHOTOGRAPHY • ROBERT LORENZ

TABLE 1: *Microprocessor Clock Rates and Access Times*

CPU	CLOCK RATE (MHz)	CLOCK CYCLE (ns)	ZERO-WAIT-STATE MEMORY ACCESS (ns)
8088/8086	4.77	210	840
8088/8086	8	500	125
8086	10	100	400
80286	6	166.6	333.3
80286	8	125	250
80286	10	100	200
80286	12	83.3	166.6
80386	16	62.5	125.5
80386	20	50	100

Memory access time depends on the microprocessor and its clock rate. The 8088/8086 processor require a minimum of 4 clock cycles to complete a memory access; both 80286 and 80386 processors require a minimum of 2 clock cycles.

TABLE 2: *Standard Machine Memory Access Characteristics*

MACHINE	CPU (MHz)	CLOCK RATE (MHz)	WAIT STATES	ACCESS SIZE
IBM PC/XT	8088	4.77	0	Byte
IBM PS/2 Model 30	8086	8	0	Word
IBM PC/AT Model 339	80286	8	1	Word
IBM PS/2 Models 50,60	80286	10	1	Word
AST Premium/286	80286	10	0	Word
IBM PS/2 Model 80	80386	16	1	DWord
Compaq Deskpro 386	80386	16	0/2[a]	DWord
PC's Limited 386[16]	80386	16	0	DWord
Compaq Deskpro 386/20	80386	20	0/4[b]	DWord
IBM PS/2 Model 80-111	80386	20	?	DWord

[a] *The Deskpro 386/16 achieves zero wait states when consecutive accesses occur within a 2KB page. Accesses outside a 2KB page occur at two wait states.*
[b] *The Deskpro 386/20 achieves zero wait states when accessing locations contained in the cache. Accesses outside the cache occur at four wait states.*

Machines that contain microprocessors running at 16-MHz and higher clock rates achieve zero-wait-state memory access using DRAM memory coupled with a high-speed SRAM cache, simple SRAM, or page-mode (static-column) RAM.

speed requirement. (Even the 386 becomes processor bound when continually accessing zero-wait-state memory, occasionally inserting idle states to enable code execution to keep up with memory accesses.) But with faster processors, providing zero-wait-state memory in a directly connected DRAM system is very expensive; and no commercially available DRAMs can keep up with the very fastest processors. Designers have used innovative memory architectures to balance high performance with reasonable cost.

In early PC systems, memory architectures were all basically the same—a simple DRAM architecture. But high-powered 386-based systems use several different kinds. These memory architectures provide improved system performance, but do they cause any software compatibility problems?

Not really. Memory subsystems are implemented at hardware level and installed on system boards of personal computers. They are transparent to software and complete when users buy their systems. If any standard is needed, it is a 32-bit bus standard for AT-style machines. Although many manufacturers offer the ability to add 32-bit memory cards to their 386 systems, no memory card is compatible with other machines. Because of this, 32-bit memory boards for a system usually must be purchased from the system's vendor at a premium price.

MEMORY ARCHITECTURE CHOICES

Several architectures are used by system designers today: simple DRAM, simple SRAM, interleaved RAM, and page-mode (or static-column) RAM. Designers are also turning to memory caching. Types of memory caching include fully associative, direct-mapped, and two-way associative caching.

Simple DRAM. A simple DRAM memory system, directly connected to the CPU, is the basic architecture that has traditionally been used in microcomputer systems. The simplest system to design, simple DRAM is the least expensive choice because it uses widely available off-the-shelf parts. However, performance of a simple DRAM architecture does not meet the needs of processors that operate at 16 MHz and faster.

In a simple DRAM system, the processor requests data from a certain location in memory. The memory system makes sure appropriate RAM chips are precharged (ready to transmit), and sends data to the processor.

DRAMs are simple semiconductor memory devices. Each memory device, or chip, contains thousands of bits (such as the 256-kilobit, or 256Kbit, chip) or even a million bits (the 1-megabit, or 1Mbit, chip). Each bit is stored in a memory cell; the memory cell consists of a capacitor element that stores an electrical charge and a transistor element that acts as a switchable connection to a data read/write line.

Because of current leakage from capacitor elements, the processor must refresh memory every 4 ms to keep memory from losing data it is storing. Refresh cycles typically do not require the processor, but the processor cannot access DRAM during a refresh cycle. In a PC, refresh is approximately a 7-percent overhead. Therefore, the processor cannot access DRAM roughly 7 percent of the time.

Although the simple DRAM architecture works fine with slower processors, DRAM chips cannot respond fast enough to provide a truly high-performance system. For example, a DRAM with an access time of 200 ns takes that amount of time to send data, once requested. (Access time is the number most frequently quoted with respect to memory chip performance, and is usually indicated in the part number on the chip package; for example, a chip with a number ending in -15 has an access time of 150 ns.)

However, DRAM chips also require precharging between accesses to get them ready for the next access. This precharge time can be just as long as the access time. The sum of access time and precharge time is called DRAM cycle time, the actual time necessary to obtain information from the chip. In 200-ns DRAMs, cycle time is 345 ns. Therefore, consecutive accesses to 200-ns memory involve not only the 200 ns required to send data, but also

a 145-ns precharge time. The 100-ns DRAMs used in the 16-MHz IBM PS/2 Model 80 have a cycle time of 187.5 ns, making them a little too slow to provide zero-wait-state memory for the 16-MHz 386 (which expects zero-wait-state access to occur in 125 ns).

The fastest DRAM chips readily available today have an access time of 80 ns. Because precharge time is also involved, a 386 system operating at 16 MHz requires one wait state even when using this memory. According to Compaq, on average the added wait state reduces system performance to 80 percent of a zero-wait-state system. A 20-MHz processor would require at least two wait states, reducing performance to a level that most designers would find totally unacceptable.

Although a simple DRAM system is the least costly memory architecture, using the fastest DRAM chips increases cost. Table 3 lists OEM prices of 64Kbit, 256Kbit, and 1Mbit DRAM chips of various speeds.

As a practical matter, DRAM chips must be used with control circuitry designed to take advantage of their best performance. A system's performance cannot be improved simply by replacing its memory chips with ones with faster access time ratings. Installing 80-ns DRAMs in an 8-MHz AT does no good unless they are in a memory board that is configured to process memory accesses at zero wait states (such as the Cheetah Card from Cheetah International, Inc.)

Simple SRAM. A simple SRAM architecture is on the other ends of both the performance and price spectra from simple DRAM architecture. It offers the highest performance, but at the highest cost. SRAM chips are available with access times as low as 15 ns, but they are considerably more expensive than DRAM chips. SRAM chips contain memory cells consisting of a "flip-flop" circuit made up of two sets of cross-coupled electrical components.

Because SRAMs contain more electrical elements per bit, they have a lower packing density. Thus they have a higher cost per bit than DRAMs, and take up more board space than DRAMs for the same amount of memory.

Functionally, a simple SRAM system is just like a simple DRAM system. When the processor requests memory, the memory system accesses appropriate memory chips and returns the requested information. However, the design of SRAM chips enables them to be produced with much shorter access times. In addition, SRAM chips can

TABLE 3: ***OEM Chip Prices***

TYPE	ACCESS TIME (ns)	SIZE (bit)	PRICE[a]
DRAM	200	64K	$ 1
	150	256K	3
	120	256K	4
	100	256K	5
	80	256K	5
	120	1M	20
	100	1M	25
Page mode	100	256K	7
SRAM	35	16K	4

[a] *Prices current as of fourth quarter 1987.*

Memory prices are directly related to performance. Low-speed DRAM is the least expensive per Kbit, high-speed SRAM is the most expensive, and page-mode (static-column) RAM is between both in performance and cost.

maintain their information without constant refreshing by the processor.

Overall system performance can increase because the system does not need to insert refresh cycles periodically, as it does with any system that uses DRAM chips. However, if users insert memory cards containing DRAMs into expansion slots of the computer, refresh cycles will be needed, negating some of the performance improvement provided by the SRAM system.

The higher performance of a simple SRAM system costs extra dollars, which is why few systems use the simple SRAM architecture. The most popular system using the simple SRAM architecture is the PCs Limited 386[16].

Interleaved RAM. Interleaved, or bank-switched, RAM architecture uses ordinary DRAM chips, but improves performance by dividing memory into banks and overlapping the precharge time in one bank with memory access in another bank. This scheme assumes that most memory accesses are sequentially ordered. Therefore, memory is divided into two or more banks, with sequential addresses interleaved among the banks. For example, in a 32-bit computer with two memory banks, the first 32 bits would be in bank 0, the second 32 bits in bank 1, the third 32 bits back in bank 0, and so on. Figure 1 illustrates interleaved architecture.

Interleaving speeds sequential accesses by enabling one memory bank to be preparing a subsequent 32-bit word for the processor while the processor accesses the current 32-bit word from the other bank. However, the benefit of overlapped accesses is not without penalty. Additional logic is required to implement interleaved memory, but this logic is not as complicated as caching logic nor as expensive as SRAM or page-mode RAM chips. Plus, initial accesses (and consecutive accesses to the same bank) are longer than they would be if interleaving circuitry were not present.

Interleaving improves performance of sequential accesses, but memory accesses often bounce around between fetching code and data items (which may or may not be in the same bank) with few extended periods of sequential accesses. In situations like this, much of the advantage of an interleaved architecture is lost.

Another disadvantage to interleaving is that the increments in which memory can be added to a system are double what they are for a noninterleaved system. For example, in a 386 system using 256Kbit chips and memory that is 32 bits wide, memory could be added in 1MB increments for a noninterleaved system, whereas an interleaved system, with its two banks, requires 2MB increments.

Interleaved RAM is used on some earlier 386 systems, mainly those using Intel's iSBC 386AT motherboard such as the ALR Access 386.

Page-mode RAM. Like interleaved RAM, page-mode RAM architecture is one that improves overall access time of DRAMs by offering higher performance for certain memory accesses. Unlike interleaved RAM, page-mode RAM (also known as static-column RAM) requires a special DRAM chip that is more expensive and harder to find than ordinary DRAM chips. It also requires special circuitry to take advantage of those chips. However, performance of systems using page-mode DRAM is usually better overall than systems using interleaved DRAM.

With page-mode RAM, system RAM is divided into equally sized areas, called pages. With paging circuitry in place, successive accesses to memory within a page are extremely fast, while successive accesses to different pages are somewhat slower. Frequent alternation between code and data memory accesses when code and data are in different pages negates some advantages of paging. Accessing memory that is outside a page boundary enacts the same sort of performance penalty as consecutively accessing a single bank of an interleaved system.

On 386-based computers that use memory paging, each bank of RAM (usually 1MB or 4MB in length) is logically divided into rows and columns.

ALL GAIN, NO PAIN

Blow away the 640K barrier

Gain the benefits of protected mode the easy way with OS/286™ and OS/386™. These tools for C, Fortran, Pascal and Assembly language programmers permit rapid conversion of existing DOS applications from "real" 8086 mode to "protected" 286 and 386 mode. They don't replace or modify DOS, but extend it to protected mode.

OS/286 and OS/386 are the only DOS extenders that span both the 286 and 386 processors, with 32-bit capability *today* on 386s that yields twice the performance of 16-bit mode. OS/286 and OS/386 have quickly become the preferred solution for developers of high performance, memory-intensive applications, including CADKEY, CASE, and Gold Hill, and premier language developers Lahey, Metaware and Green Hills.

Our optional TOUCHDOWN™ BIOS supplement provides fast and reliable protected mode operation on *any* 286 system, even those with problems resetting the 286. (Ever notice how few existing machines Operating System/2 runs on?)

If your applications are running out of memory or need more speed, don't wait for the "solution" that means abandoning your investment in DOS. Enhance them now with OS/286 and OS/386 — *products not promises*.

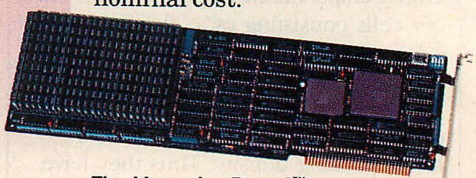
OS/286™ & OS/386™ Benefits:

- Gain multi-megabytes of directly addressable memory (15-Mb-286, 4Gb-386)
- Increase performance by *eliminating* overlays and EMS
- Convert in days, not months
- Continue to work with a DOS interface and use existing TSRs, device drivers, graphic routines, etc.
- Stay compatible with the widest array of systems

A.I. Architects Software Developers Kit **$495** includes full support for:

- MetaWare High C (16 & 32 bit)
- Professional Pascal (16 & 32 bit)
- Lahey F77L FORTRAN
- Microsoft C & Fortran 4.0,
- MASM, MS-Link
- Phoenix PLINK86
- Halo & GSS Graphics
- Pharlap 386: ASM/LINK more to come

Run time licenses for OS/286 and OS/386 are available at nominal cost.



The HummingBoard™ turns any XT or AT into the fastest 386 system available. The dual processor architecture boosts performance significantly over comparable single processor systems or accelerator boards. Available with 2 to 24Mb RAM, 16 or 20Mhz speed, and 387 floating point coprocessor.

A.I. Architects, Inc.
One Kendall Square, Building 400
Cambridge, Massachusetts 02139
(617) 577-8052

CIRCLE NO. 253 ON READER SERVICE CARD

Access to memory is obtained by presenting a row address and column address. These two addresses are latched internally using the row address strobe (RAS) and column address strobe (CAS). The RAS and CAS combination identifies a specific 32-bit word, the amount of memory that the 386 can access in a single memory fetch.

In the scheme used in the 16-MHz Compaq Deskpro 386 (which uses 1MB memory banks), each row and column address is a 9-bit quantity, implying 2^9 (or 512) items in each row and column. With each item being 4 bytes (32 bits) long, each page (or specific row address) of memory consists of 512 (2^9 column addresses) by 4 bytes, or 2KB of memory. The 512 pages enable the addressing scheme to handle 1MB of memory. Additional 1MB banks are accessed in the same manner, with their own row and column addresses. Figure 2 illustrates row and column addresses.

With page-mode DRAM chips, memory accesses within the same page can happen very quickly (at zero wait states with a 16-MHz 386) because RAS maintains the row address from the previous memory access. When a new memory access is required, the row address is compared with the value already maintained by the memory chips.

If the new row address is the same as the previous one (that is, the memory is in the same 2KB page), only a new column address needs to be presented to the DRAM. This results in a fast access. However, if the new row address is different (that is, the memory is in a different 2KB bank), the CPU must wait for memory to precharge and present a new row address followed by a column address, resulting in a two-wait-state access.

The Compaq Deskpro 386/16 and Portable 386 use a page-mode DRAM in its memory architecture. Because most personal computer applications average 60 percent of their memory accesses within a 2KB page, these machines operate at approximately .8 wait states per memory access.

As a practical matter, users purchase memory expansion kits the computer vendor or a third party provides for their computers. Table 4 shows the cost of memory expansion kits for several computers, type of memory used, and cost per megabyte of the particular memory expansion option.

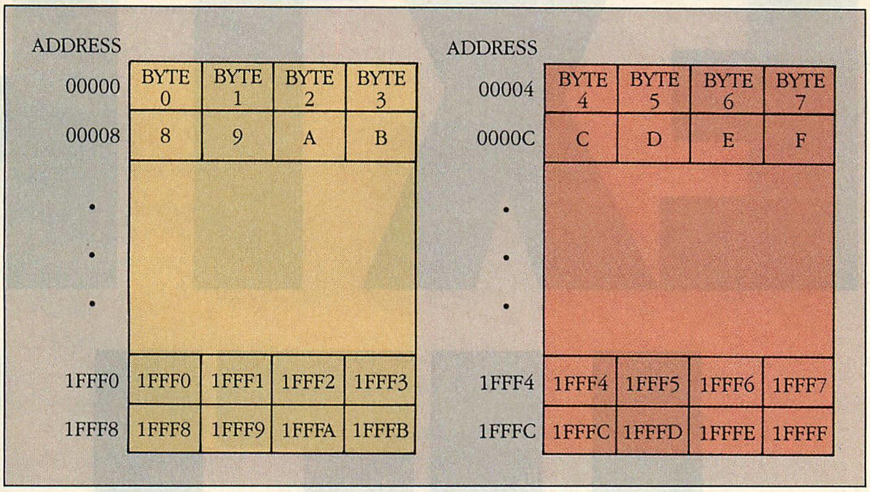
CACHING IN THE CHIPS

A cache memory architecture combines SRAM's speed with DRAM's cost effectiveness. It provides a small amount (usually 64KB or less) of fast SRAM (the cache) that is logically located between the processor and main memory (which is usually simple DRAM, although some systems improve performance even more by using interleaved or paged memory for their main memory). SRAM in the cache usually has an access time of 35 ns or less, enabling even 20-MHz 386s to access data in the cache at zero wait states. Cache circuitry makes sure that the most-often-used portions of main memory are copied into the cache. Therefore, the majority of the memory accesses will be to fast memory in the cache, not to slower main memory.

A cache memory system is relatively expensive because it requires SRAM chips for the cache and complex cache control circuitry to regulate the operation of the cache. However, sophisticated cache controller chips, such as the Intel 82385 (see the accompanying sidebar on page 92), can provide the entire controlling logic on a single chip. Table 3 lists the cost of SRAM chips used in cache memory.

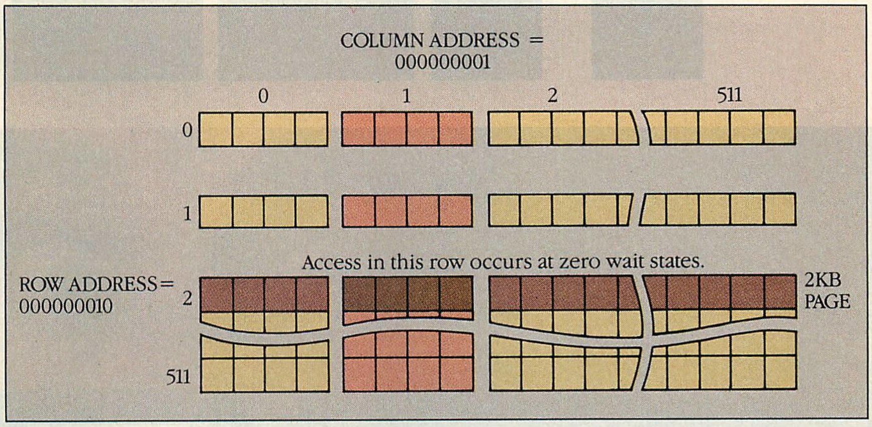
In a system that uses a cache, whenever the processor attempts to read a memory location, the memory subsystem checks to see if the contents of that location are stored in the cache. If so, the data are transferred from the cache at fast, SRAM speed. If the data are not in the cache, the processor must wait until the data can be transferred from slower main memory. At the same time, the contents of the location are also copied into the cache, where they can be accessed more quickly the next time they are needed.

FIGURE 1: ***Interleaved Architecture***



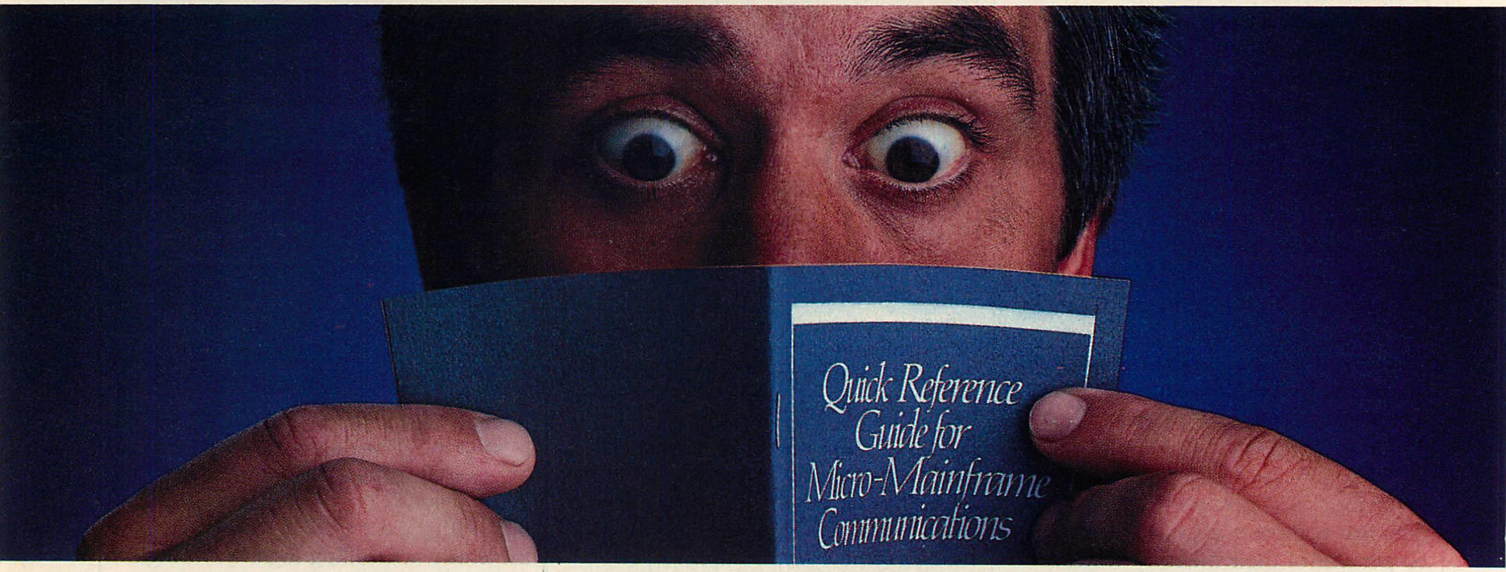
In this architecture, memory addresses are divided into two banks, with addresses interleaved between the banks. During sequential accesses, one bank can be precharged while the other one is being accessed. This figure shows two 1MB banks.

FIGURE 2: ***Page-mode Memory Access***



To access memory in each 1MB bank, the memory subsystem specifies 9-bit row and column addresses. Page-mode RAM allows the memory to respond in zero wait states if the row address is the same as in the previous memory access.

EXTRA HELP



There's a remarkable world of micro-mainframe communications that lies beyond basic 3270 emulation. Attachmate will give you the extra help you need to discover it.

Our new software—Extra! Connectivity Software™ —is the most powerful connectivity software available for the IBM® environment, including the new PS/2™.

Our popular *Quick Reference Guide for Micro-Mainframe Communications* compares IBM, IRMA,™ and EXTRA! and answers important questions about the new IBM standards, multiple sessions, windows, file transfer, API and graphics. Information you should know before you make long-term commitments.

If you'd like a copy of our free guide, connect with Attachmate today. Call toll free:

1-800-426-6283

(1-800-IBM-MATE)

Attachmate

Quality Micro-Mainframe Solutions

Attachmate Corporation
3241 118th S.E., Bellevue, WA 98005
(206) 644-4010

TABLE 4: ***Memory Expansion Options***

MACHINE	TYPE OF MEMORY	PATH WIDTH (BITS)	FULL BOARD SIZE	PRICE	COST PER MEGABYTE
IBM PC/XT	DRAM	8	256KB[a]	$ 60	$240
IBM PC/AT	DRAM	16	2MB	750	375
IBM PS/2 Models 50/60	DRAM	16	2MB	1,044	522
ALR Access 386	Interleaved	32	2MB	1,350	675
IBM PS/2 Model 80	DRAM	32	6MB	4,185	698
Compaq Deskpro 386	Static column	32	8MB[a]	4,198	525
PCs Limited 386[16]	SRAM	32	1MB[a]	499	499
Compaq Deskpro 386/20	DRAM	32	4MB[a]	2,099	525

[a] Memory expansion kit that is added to system/system-memory board.

Most 386 machines use more complex types of memory, whereas lower-performance machines use simple DRAM. The Compaq Deskpro 386/20 uses DRAM, yet still provides high performance due to its use of a high-speed SRAM cache.

A cache is made effective by the tendency of most computer programs to access the same few memory locations over and over, and the tendency to access neighboring locations of those accessed recently. With a cache architecture, once those few locations have been loaded into the cache, system performance increases because most accesses are from the cache, not from slower main memory.

One measure of a cache's effectiveness is the ratio of accesses in the cache compared to total memory accesses. This ratio is called the *hit rate*. For example, if 90 percent of all memory accesses are to locations already stored in the cache, the cache's hit rate is 90 percent. An effective cache requires a hit rate of 60 percent or more. A high hit rate is usually necessary because even though accesses to locations in the cache are very fast, accesses to locations outside the cache are slightly slower than they would be without the cache present. The extra time is necessary to update the cache with the contents of the new location.

The size of the cache, the size of the block, and the design of the cache all affect the cache's hit rate.

Cache size. The simplest way to increase the hit rate is to increase the cache's size. The larger the cache, the more locations it can store, and therefore the greater the chance that a given location will already be stored in the cache when needed.

The minimum cache size necessary to achieve a 50- to 60-percent hit rate is 2KB. More typical sizes for 386-based systems is are 32KB and 64KB. Orchid Technology's Jet 386 accelerator board uses a 64KB cache, while the Compaq Deskpro 386/20 uses a 32KB cache.

Of course, the cache size refers only to the SRAM that is used to store the actual contents of the memory locations. Additional SRAM is required to keep track of the locations stored in the cache. The amount of this memory, which is called tag RAM, depends on the design of the cache. New cache controllers, such as the Intel 82385, provide both the control circuitry and tag RAM in a single chip.

Block size. The block size can affect the hit rate of the cache. The block size is the amount of data transferred to the cache each time the cache needs to be updated. Typical block sizes are 2, 4, 8, 16, or 32 bytes. Cache systems for the 32-bit 386 often use a block size of at least four bytes. Whenever a word that has been requested by the processor is not in the cache, the cache controller copies not just that word but the entire block containing that word into the cache.

A cache that uses a large block size has fewer blocks, therefore increasing the danger of a block being overwritten soon after it is fetched. In addition, in a 386 system, blocks larger than 32 bits require multiple memory accesses, making each update of the cache take more time.

Likewise, block sizes that are too small can greatly affect performance. Because accesses are generally to nearby areas of memory, having a small block size can minimize the effects of look-ahead and look-behind algorithms used by some caches.

Cache associativity. When the cache becomes full and the processor reads a location that is not currently stored in the cache, the memory subsystem must overwrite one of the cache locations with the location that was just read. The location that gets overwritten depends on the cache's associativity, which determines how many places one particular main memory location can be mapped in the cache.

The most flexible but most complex organization is the *fully associative cache*. In this organization, when a new block is read into the cache, the block can be placed anywhere in the cache. If the cache is full, an algorithm is used to determine which block in the cache is least likely to be needed and that particular block is overwritten by the new block. Usually a least-recently-used algorithm is applied to determine where to put the new block.

Because locations can be stored anywhere in the cache, no relationship exists between the blocks' addresses in the cache. Therefore, tag RAMs must maintain the complete address of each block in the cache. When a memory access occurs, the memory subsystem must compare the address requested with the complete list to determine whether that location is available in the cache. This means that more tag RAM is required, and therefore the fully associative cache may be slower than other kinds of caches. However, because the cache architecture is very flexible, a small, fully associative cache can provide an equivalent hit rate to a larger cache of another design.

Figure 3 illustrates a fully associative 4KB cache that can handle the full 4GB of data in the 386's address space. Each entry in the cache is four bytes long, with 1,024 distinct entries. To handle these entries, the cache must have 1,024 tag RAM locations to store the addresses of these locations. Because the entire 4GB address space can be cached, each tag RAM entry must be 30 bits. (The two low-order bits in a 32-bit address are used to select one of the bytes in a four-byte entry. Because the block size is four bytes, tag RAM does not need to store these last two bits.) Therefore, 30,720 bits, or 3,840 bytes of tag RAM are necessary just to keep track of addresses. Additional memory also is required to indicate whether the data contained in the block are valid or invalid.

On the other extreme from the fully associative cache is the *direct-mapped cache*. Each location in main system memory can be copied to one and only one location in a direct-mapped cache. For example, with a 32KB cache, all of the main memory is divided into 32KB pages. The first location of any 32KB page, when copied to the cache, occupies the first location in

the cache. Likewise, the ninth location in each 32KB page occupies the ninth entry in the cache.

Organizing a cache in this manner involves less complicated cache circuitry and may be faster because, whenever a memory access occurs, the cache controller must check only one entry in tag RAM to determine whether that location is present in the cache.

Figure 4 illustrates a 32KB direct-mapped cache that can handle the full 4GB address space of the 386. Each entry in the cache is four bytes long, with 8K (8,192) entries in the cache. To handle these entries the tag RAM must have 8,192 entries to store addresses. However, unlike fully associative architecture, each tag RAM entry for a direct-mapped cache needs to be only 17 bits long. The tag RAM entry chooses one of the 131,072 (2^{17}) 32KB pages of memory. The remaining 15 bits of a 32-bit address are an index into tag RAM entries (the upper 13 bits) and an indication of which byte of the four-byte cache entry is being accessed.

Although a direct-mapped cache involves relatively simple hardware, its major disadvantage is that each location in main memory corresponds to only a single entry in the cache. Therefore, if a program constantly accesses two memory locations that correspond to the same cache entry (for example, the first word of two different 32KB pages), the cache will be ineffective because it will be constantly reloading that cache entry with the contents of one or the other location.

A *two-way associative cache* eliminates the problem of constant reloading of two locations that use the same cache entry. The Compaq Deskpro 386/20, which uses the 82385 cache controller, sets up its cache as a two-way associative cache.

Whereas a 32KB direct-mapped cache is divided into a single bank of 8,192 four-byte entries, a 32KB two-way associative cache is divided into two banks of 4K (4,096) four-byte entries. Instead of one set of tag RAMs, there are two, so each location in main memory has the possibility of being in one of two places in the cache. It is essentially the combination of two half-sized, direct-mapped caches. A two-way associative cache offers a higher hit rate than a direct-mapped cache.

Figure 5 illustrates a 32KB two-way associative cache that can handle 4GB of main memory. Because the cache is divided into two parts, each part contains 4,096 four-byte entries. Therefore, each page is 16KB long, as opposed to 32KB pages in the direct-mapped cache. This means that a 4GB memory space contains 262,144 (2^{18}) 16KB pages, therefore, each tag RAM entry must be 18 bits long.

With a two-way associative cache, the high-order 18 bits of a 32-bit memory address are the tag value, which chooses the page. The lower 14 bits

CONTROLLING THE CACHE

Intel Corporation's 82385 32-bit cache memory controller is a key component of the 20-MHz 80386 computing engine. This controller, along with the 80387 math coprocessor and 82380 integrated systems peripheral, provides fast memory access, mathematical processing, and I/O services that complement the high-speed operation of the 386.

The 82385 provides the 386 with access to system memory via a high-speed cache. The 82385 contains a cache directory (tag RAM that identifies the memory locations stored in the cache) and the circuitry to control 32KB of external cache.

The 82385's connection to the 386 and the system bus is transparent, providing access to all code and data stored in system memory. The entire 4GB physical memory space of the 80386 is mapped into the cache. The 386's local bus connects to the 82385, and the 82385's local bus connects to the system bus. The 386 operates as if it were connected to a very fast system bus. The 82385 uses an independent timing control unit, allowing the system bus to operate as if it were connected to a low-speed 386.

When the 386 initiates a memory read, the 82385 compares the high-order bits of the address supplied with the appropriate address tags stored in its cache directory. If the contents of the requested location are stored in the cache (a *read hit*), they are provided to the 386, and the 386 memory access cycle is completed without any wait states inserted.

If the contents of the memory location requested by the 386 are not stored in the cache (a *read miss*), the 82385 reads the requested location from memory, provides it to the 386, and copies it to the cache; it also updates the on-chip cache directory so that subsequent requests by the 386 for the location will occur with zero wait states inserted.

When the 82385 performs a memory read in response to a read miss, it provides one full 32-bit word to the 386, as well as copying it to the cache. It does not prefetch additional data from memory.

Memory writes from the 386 are handled by the 82385 on a posted-write-through basis. Data from the 386 are always written to system memory. If a copy of the memory location being written to memory is stored in the cache, it is updated; otherwise, the cache is unchanged. Writes to system memory are buffered; 386 memory writes occur with zero wait states inserted, provided the previous 386 write has been completed on the system bus.

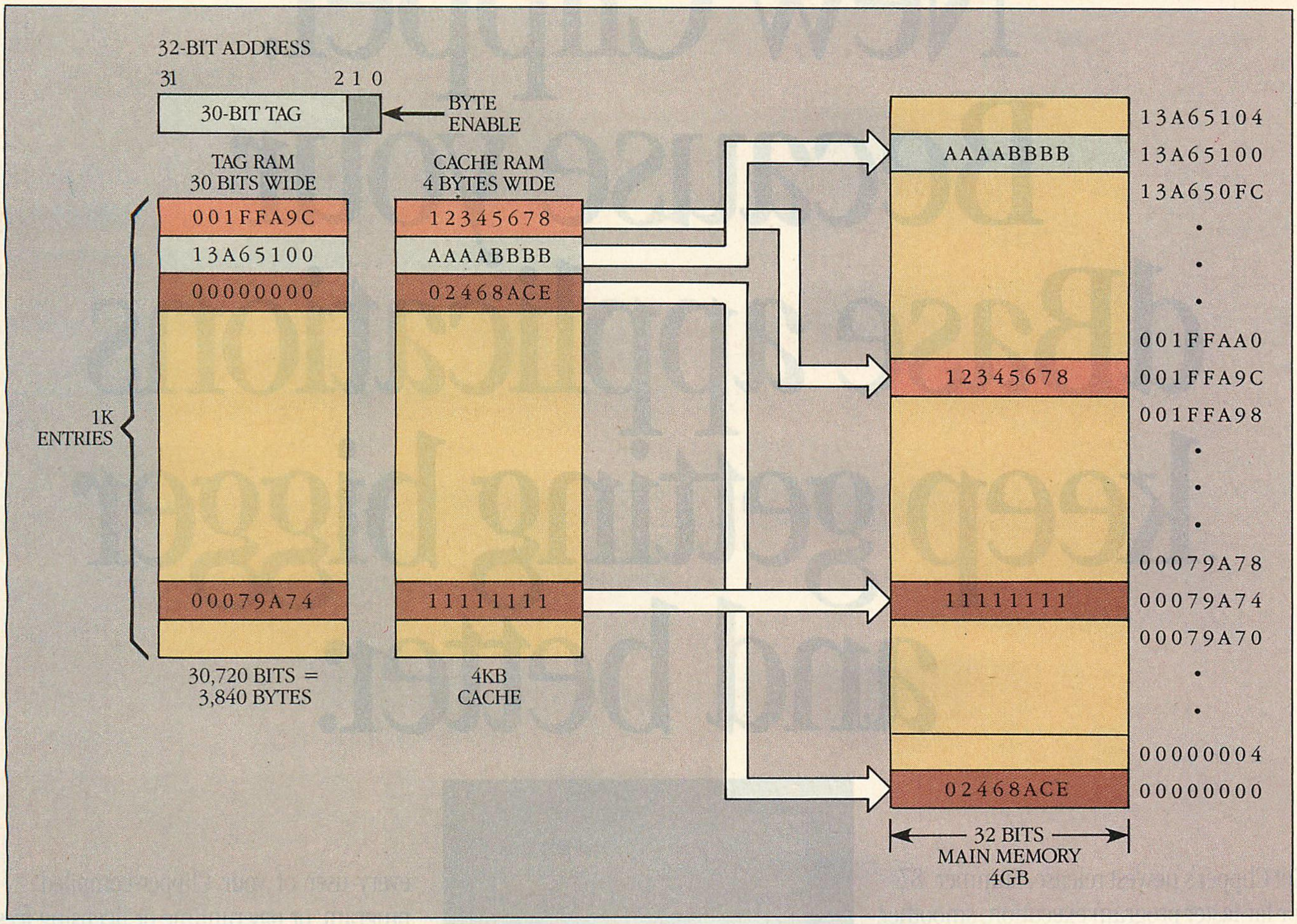
The 82385 assures that the contents of the cache accurately reflect the contents of system memory by monitoring activity on the system bus whenever it is not itself controlling the bus. If another bus master writes to a location in main memory, the 82385 determines if that location is stored in the cache, and if so, invalidates its entry in the cache directory. Bus monitoring can be performed every other clock cycle without slowing the 386's access to the cache because a 386 memory access requires a minimum of two clock cycles, and cache hit/miss determination requires only one clock cycle.

The 82385 can be configured to operate as a direct-mapped or two-way-associative cache via a strapping option. When configured for direct-mapped operation, the 32KB cache is structured as 8K 32-bit words, and system memory is viewed as contiguous 32KB pages. The results of a memory read from a 32-bit word location in a 32KB system memory page are stored in the corresponding 32-bit word in the cache.

When configured for two-way-associative operation, the cache is structured as two banks of 4K 32-bit words each, and system memory is viewed as consisting of contiguous 16KB pages. When a read miss occurs, the 32-bit word location is read from the appropriate 16KB system memory page and stored in the least recently used of the two corresponding 32-bit words.

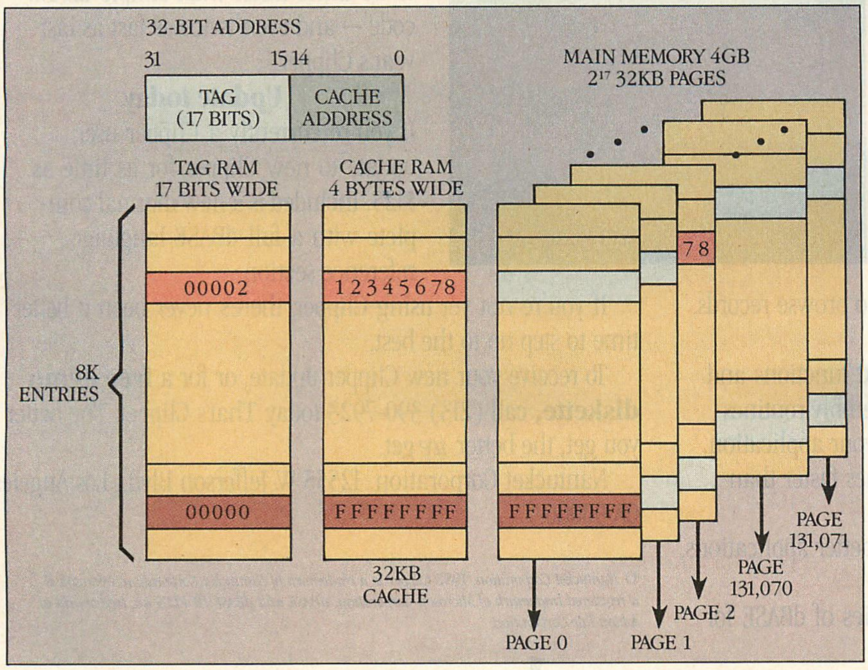
—JS

FIGURE 3: *Fully Associative 4KB Cache*



Each cache block is 4 bytes long, so there are 1,024 blocks contained in the cache. Contents of locations in the 4GB main memory can be stored in the cache in no particular order. The tag RAM stores the starting address of each 4-byte block.

FIGURE 4: *Direct-mapped 32KB Cache*



Main system memory is divided into 131,072 32KB pages. Locations are mapped to the cache in the same position that they occupy in the main memory page.

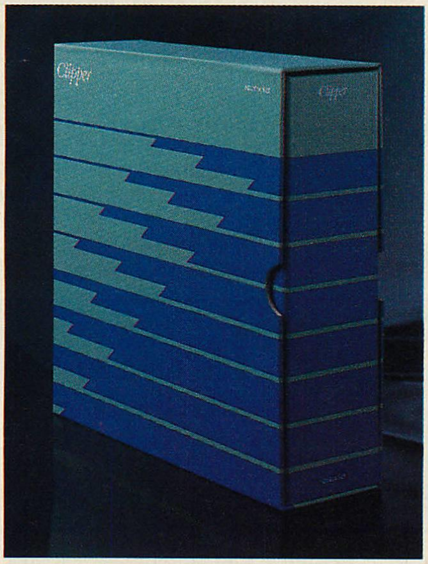
are an index into two separate lists of tag RAM entries (the upper 12 bits) and indicate which byte of the four-byte cache entry is being accessed. Because the address indexes into two sets of tag RAMs, any entry has two chances of being in the cache.

Cache associativity can be implemented in other organizations, such as *three-way* and *four-way associative caches*. The principles are the same, but the cache and tag RAM are divided into smaller pieces. Direct and two-way caches, however, are destined to become the most used because the popular Intel 82385 cache controller supports both of these organizations.

WRITING DATA TO MEMORY

Most of the discussion of memory accesses via a cache apply to reading information, not writing it. How the cache is used when writing information to memory depends on the cache controller's sophistication. Simpler caches use a write-through policy in which every write access causes the system to write to main memory, whether or not

New Clipper.™ Because your dBase applications keep getting bigger and better.



Get Clipper's newest release, Summer '87, for far faster program execution, smoother, speedier development cycles and faster compiling than ever before.

The more sophisticated your database applications become, the more you need the Clipper compiler.

More Programming Power.
New commands, functions and dBASE® compatible indexing make new Clipper the most powerful solution yet for creating the types of dBASE III PLUS™ applications you're working on today. Single-user or networking. For example, fully programmable functions make it easy to browse records, create pop-up menus or edit free text.

It's easy to develop your own user-defined functions and libraries with the Microsoft® C compiler, assembly routines or Clipper itself, and incorporate them into your application. And new Clipper compiles your code five times faster than its predecessor.

Bottom line: You get more sophisticated, better applications, written quicker.

And you *still* don't have to buy extra copies of dBASE for every user of your Clipper-compiled program, or pay runtime or licensing fees.

Faster Program Execution.
dBASE can bog down in complex applications. Not with new Clipper. It's 10 to 20 times faster than simple dBASE code — and even twice as fast as last year's Clipper.

Update today.
If you're currently a Clipper user, update to new Clipper for as little as $125. Included is a new manual complete with a full dBASE language reference section.

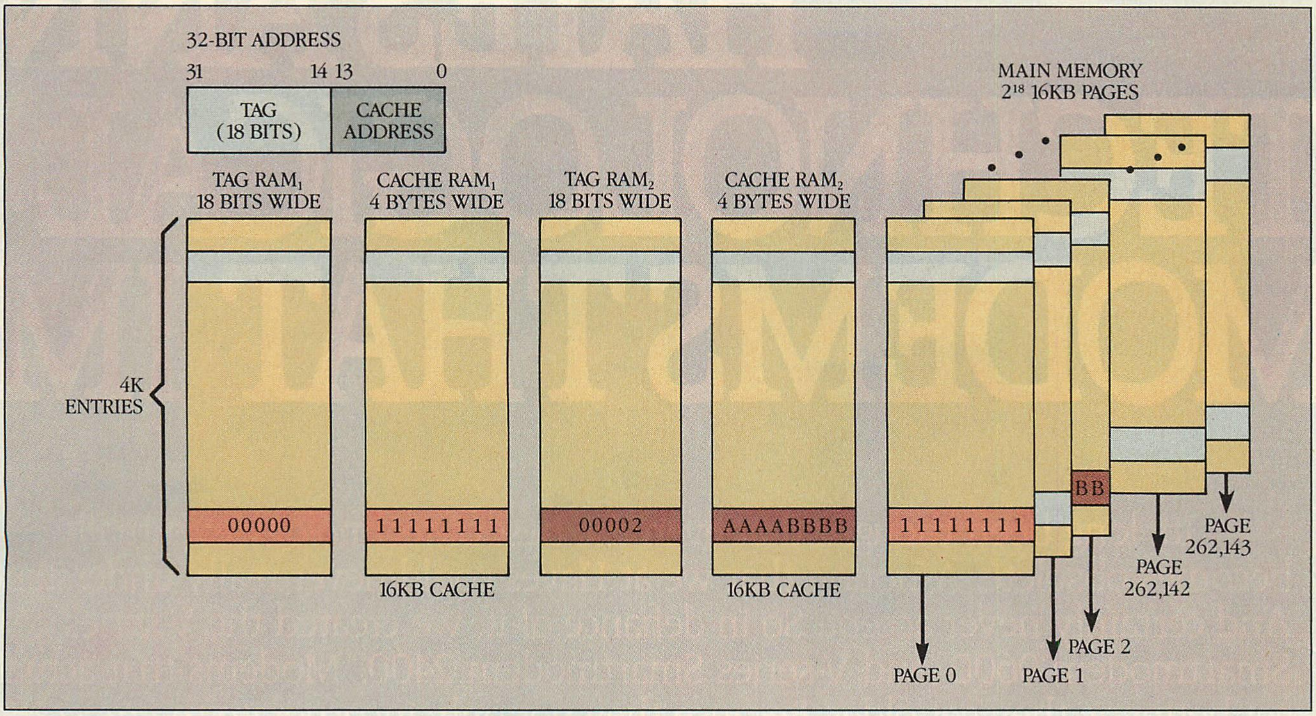
If you're not yet using Clipper, there's never been a better time to step up to the best.

To receive your new Clipper update, or for a **free demo diskette,** call (213) 390-7923 today. That's Clipper. The better you get, the better *we* get.

Nantucket Corporation. 12555 W. Jefferson Blvd., Los Angeles, CA 90066.

Nantucket®

FIGURE 5: ***Two-way Associative 32KB Cache***



This 32KB cache is split into two 16KB direct-mapped caches. The page size is cut to 16KB, and the number of pages in the 4GB main memory is doubled to 262,144. When a location is mapped to the cache, it occupies a position relative to that in the main memory page. However, because the cache is split, any main memory location can occupy one of two positions.

the affected location is stored in the cache. This can cause performance degradation in applications that write to memory a lot, because these accesses are not taking advantage of the cache.

Sophisticated cache controllers, such as the 82385, allow posted write operations using the cache. If the location to be written is in the cache, the cache is updated at zero wait states. Then, while the CPU continues executing program code, the cache controller simultaneously and automatically handles transferring data from the cache to main memory. Because 10 to 15 percent of all memory accesses are writes in most business applications, this ability to support posted write operations can improve overall system performance significantly.

Cache systems that use posted write operations can have problems with data latency, however, because there is a short period of time when information a processor writes to memory is in the cache, but the cache has not yet updated main memory. If another operation that reads main memory, such as direct memory access (DMA), occurs during this time, it will retrieve outdated information.

Another area affecting memory performance is how the cache operates when simultaneous memory writes are occurring. For example, many systems allow devices to perform DMA transfers to memory while the CPU continues to execute code. Some systems flush the cache when another bus master writes to main memory. This slows memory accesses because the cache will have to be loaded again before zero-wait-state operation can continue. Other systems route all DMA transfers through the cache, but this ties up the system bus and also slows overall system performance. Cache controllers such as the Intel 82385 handle this problem by implementing a snooping strategy. The cache controller actually detects when a DMA transfer occurs and checks each address written to determine if it is in the cache. If it is, the controller invalidates that location only, not the entire contents of the cache.

The Compaq Deskpro 386/20 uses a 32KB cache controlled by an Intel 82385 cache controller set up for two-way interleaving and posted writes. With this setup, Compaq claims a 95-percent hit rate for the cache.

BUILDING THE PERFECT BEAST

With the advent of faster and faster 32-bit microcomputers, designers must use creative techniques to squeeze maximum performance out of their systems while maintaining a reasonable price. This price/performance tradeoff is really the major issue facing system designers, and one that users should consider before purchasing new systems. Just because a low-cost computer contains a fast 386 processor does not mean the computer will perform as well as a higher-priced model. At 16 and 20 MHz, memory architecture can greatly affect overall performance.

Which architecture is likely to dominate in the future? For CPU clock rates below 16 MHz, simple DRAM architecture will continue to dominate, just as it has in the past. For these systems, the cost of implementing more complicated architectures is not matched by corresponding performance increases.

However, for clock rates of 16 MHz or greater, simple DRAM architecture is not sufficient. For systems running at 20 MHz and beyond, both simple SRAM and cache RAM systems seem to be the only ones capable of matching the speed of the CPU. Because of the lower cost and the presence of single-chip cache controllers, it seems likely that cache RAM organizations will dominate high-end systems.

Steven Armbrust is a freelance technical writer, and Ted Forgeron works as a program manager for Intel Corporation.

HAYES ANN TECHNOLOGICAL MODEMS THAT IM

It's long been thought that even the best of technology eventually becomes obsolete. A notion that we at Hayes could never really understand. And certainly never accept. So in defiance of it, we created the V-series Smartmodem 9600™ and V-series Smartmodem 2400.™ Modems that actually get better as they get older. Because they not only incorporate the most intelligent features found in modems today, they also possess the capability to provide a long-term growth path into the communications environments of tomorrow.

V-SERIES SMARTMODEM 9600

This is the fastest modem we've ever made. It can send and receive data at 9600 bps and with adaptive data compression achieve an effective throughput of up to 19,200 bps. Point-to-point error control, forward error correction and data flow control ensure that data gets there accurately.

The V-series Smartmodem 9600 also comes with automatic feature negotiation, a self-operating capability that analyzes all options for modem link and then selects the optimum feature set with any Hayes modem for the most efficient transmission at the highest shared speed.

Synchronous and asynchronous communications modes as well as simulated full-duplex employing advanced CCITT V.32 trellis code modulation and fast turnaround ping-pong technology are also part of the package. Plus you'll get the capability to link up with a range of networks, including SNA. And soon V-series technology will offer an X.25 PAD option to further accommodate network environments of today. And the future.

OUNCES A CONTRADICTION: PROVE WITH AGE.

V-SERIES SMARTMODEM 2400

With adaptive data compression this modem can achieve an effective throughput of up to 4800 bps. Of course, it too offers point-to-point error control, data flow control, automatic feature negotiation and synchronous as well as asynchronous communications modes. And like the V-series Smartmodem 9600, it can link up with a wide range of networks, such as SNA, and be enhanced with an X.25 PAD option.

V-series modems come in stand-alone versions and internal versions (V-series Smartmodem 9600B™ and V-series Smartmodem 2400B™). Internal versions are bundled with our powerful new Smartcom III™ communications software.

And as yet another rebuttal to the argument for obsolescence, we developed the V-series Modem Enhancer.™ A separate stand-alone device that will upgrade current Smartmodem 1200™ and Smartmodem 2400™ external modems to the new standards set by the V-series products.

A closer look at the V-series product line will reveal to you a revolutionary technology designed to be the beneficiary of time, not its victim. So contact your Hayes Advanced System Dealer or call **800-635-1225** for the one nearest you.

Introducing the two on earth

The new COMPAQ DESKPRO 386/20™

The world now has two new benchmarks from the leader in high-performance personal computing. The new 20-MHz COMPAQ DESKPRO 386/20 and the 20-lb., 20-MHz COMPAQ PORTABLE 386 deliver system performance that can rival minicomputers'. Plus they introduce advanced capabilities without sacrificing compatibility with the software and hardware you already own.

Both employ an industry-standard Intel® 80386 microprocessor and sophisticated 32-bit architecture. Our newest portable is up to 25% faster and our desktop is actually up to 50% faster than 16-MHz 386 PC's. But we did much more than simply increase the clock speed.

For instance, the COMPAQ DESKPRO 386/20 uses a cache memory controller. It complements the speed of the microprocessor, providing an increase in system performance up to 25% over other 20-MHz 386 PC's. It's also the first PC to offer an optional Weitek™ Coprocessor Board, which can give it the performance of a dedicated engineering workstation at a fraction of the cost.

They both provide the most storage and memory within their classes. Up to 300 MB of storage in our latest desktop and up to 100 MB in our new portable.

It simply works better.

most powerful PC's and off.

and the new 20-MHz COMPAQ PORTABLE 386™

Both use disk caching to inject more speed into disk-intensive applications and both will run MS® OS/2.

As for memory, get up to 16 MB of high-speed 32-bit RAM with the COMPAQ DESKPRO 386/20 and up to 10 MB with the COMPAQ PORTABLE 386. Both computers feature the COMPAQ® Expanded Memory Manager, which supports the Lotus®/Intel®/Microsoft® Expanded Memory Specification to break the 640-Kbyte barrier imposed by DOS.

With these new computers plus the original COMPAQ DESKPRO 386™, we now offer the broadest line of high-performance 386 solutions. They all let you run software being written to take advantage of 386 technology, including Microsoft® Windows/386 Presentation Manager. It provides multitasking capabilities with today's DOS applications to make you considerably more productive. But that's just the beginning. For more information, call 1-800-231-0900, Operator 43. In Canada, call 416-733-7876, Operator 43.

COMPAQ®

Network on a Bus

A clustered-CPU system such as Alloy's PC-PLUS is one more connectivity solution from which to choose if local area networks and multiuser operating systems do not match your needs.

GARY SKIBA

A small business manager is in the market for a connected PC-based system. He first looks at local area networks (LANs), but fears that the cost of separate PCs for every worker in addition to the network itself may be too high. He then turns his attention to a multiuser operating system, such as UNIX, but worries not only about learning a new operating system but also about compatibility with the applications he wants to use.

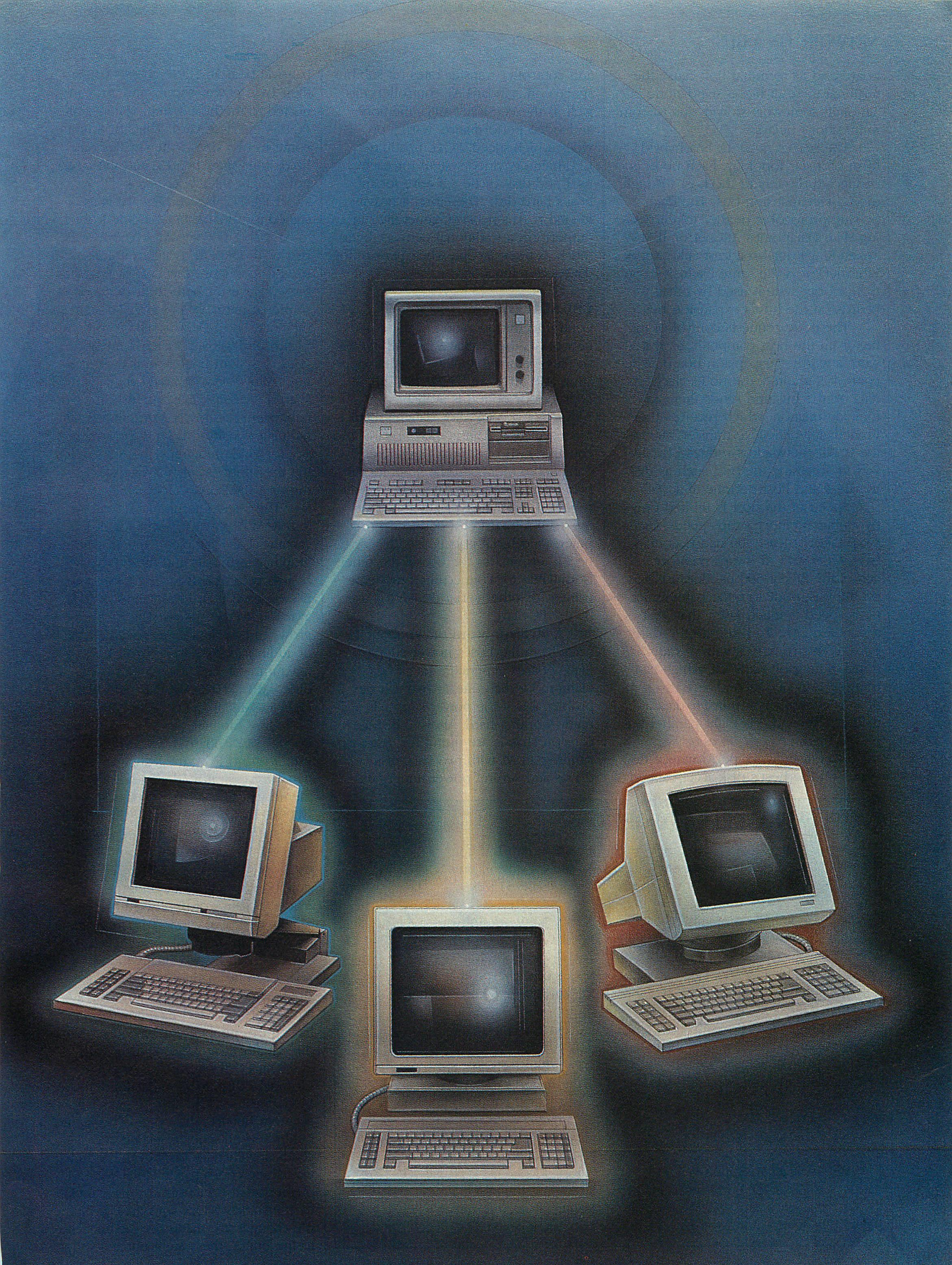
A manager in such a situation has a third alternative: a clustered-CPU system. Unlike a LAN, which consists of separate computers communicating through external hardware connections, a clustered-CPU system consists of several PC-compatible CPU boards that are physically housed in one computer; these CPUs communicate with each other (or the host) via the host computer's expansion bus.

Alloy Computer Systems Inc.'s PC-PLUS is one such clustered-CPU system, providing shared disk storage and peripherals, with a dedicated CPU for each user, *without* the expense of an entire computer for each user. First developed by Alloy in 1984, the PC-PLUS network consists of 8088-compatible CPU boards and Network Executive (NTNX) support software that together provide what Alloy calls a "network on a bus." PC-PLUS features and options are listed in the accompanying sidebar.

Alloy recently upgraded its network by introducing an 80286-compatible CPU board (see the accompanying sidebar, "Upgrading PC-PLUS to 286 Status") and by delivering a Micro Channel adapter that allows the network to run on PS/2 machines using an Alloy expansion chassis.

The PC-PLUS CPU boards, called PC-Slave/16N, reside in the host machine, which can be any IBM-compatible computer with a hard disk, from a PC/XT to an 80386-based machine. Alloy supplies optional XBUS expansion chassis that can accommodate as many as 31 PC-Slave boards in one host machine. Each board is connected to a PC-compatible terminal via a serial cable. The PC-Slave

ILLUSTRATION • ANDY LEVINE



board and its terminal link cause the terminal to react like a stand-alone PC, although somewhat slower due to communications delays.

The PC-PLUS network is a hybrid form of a star topology. Each node is connected directly to the host through the host's expansion bus. The data-passing protocol is neither token passing (like Datapoint's ARCnet or IBM's Token-Ring) nor broadcast (like Xerox's Ethernet); it is the same exclusive control priority system used by other boards on the host's expansion bus.

The node requests control of the bus from the host computer, as if it were any other expansion board. When control is granted, the node places its data on the bus and sends them to their destination, which receives them immediately. Effective transfer rate of the PC-PLUS network is reduced by delays in accessing the bus, such as disk transfers and host peripheral access.

Speed of the host is limited only by its bus memory swap speeds; each swap takes at least 400 nanoseconds (ns). An 8-MHz AT swaps memory across the bus at 500 ns. Because it is on the bus, node-to-node communications speed is much faster than that of a standard LAN. To illustrate the difference, compare PC-PLUS with a traditional, 8-megabit network, which can transfer 1MB per second. PC-PLUS on a host AT typically can transfer data at 8MB per second (via eight parallel bits on the bus), or *eight times as fast*.

Overall network performance depends on the host's disk-access speed and controller transfer rates (at sub-bus speeds on standard AT controllers) as well as the PC-Slave's CPU, memory, and video performance. PC-PLUS provides a high-speed, PC-compatible CPU/memory/video system for each user and disk cache software for the host's hard disk. Therefore, a heavily loaded PC-PLUS network generally can outperform a traditional network by as much as three to one when performing data access functions.

CPU SLAVERY

Central to the operation of the PC-PLUS network is the PC-Slave/16N, which has an 8088-compatible NEC V20 CPU operating at 8 MHz. An 8-MHz 8087 math coprocessor option is available, but requires a special Alloy daughterboard for installation. The PC-Slave board also contains 1MB of RAM, a Hercules-compatible monochrome graphics display interface, two serial ports, and all required internals typically found on the system board of a desktop computer. One serial port is used to connect a terminal to the board; the other is available for connecting a modem, printer, or other serial device for use by the PC-Slave.

A PC-Slave board draws about 15 watts from the host's power supply. Slave boards generate a lot of heat; using more than one or two can be problematic if the power supply is already working hard. Alloy's XBUS expansion chassis are useful for systems running an AT with two hard-disk drives and several power-hungry expansion boards. These chassis contain their own power supply and are connected to the host system via an 8-bit PC-bus interface card. The XBUS units come in two sizes—the smaller one holds four slave boards, and the larger holds eight slave boards.

Each PC-Slave/16N board is assigned a user number ranging from 1 to 31, using DIP switches. The host PC is user 0. The DIP switches are also used to configure the 1MB of RAM on the PC-Slave for both local DOS conventional memory and global system memory. Local DOS usually receives 512KB or 640KB; remaining memory goes to system use, such as disk caching. (Alloy says that it will soon offer a shared RAM disk that will also use global system memory.) A graphics terminal usurps half of the non-DOS memory for video functions, thereby depleting the supply available for disk caching or a RAM disk.

The system accumulates total system RAM from each board for disk caching. For example, a system with no graphics terminals and three PC-Slave boards with 512KB reserved for system use on each board has 1.5MB of RAM available to the system. The cache grows to accommodate the added load of additional PC-Slave boards, up to a maximum of 2MB for a 32MB hard disk, 4MB for a 64MB disk, and 8MB for a 128MB disk.

Alloy supplies two versions of the PC-Slave/16N board, depending on EGA use. With no EGA, the standard, *dual-window* version works fine, using memory at two segments: A000H and D000H. However, with an EGA (which uses segment A000H) the *single-window* version is necessary because it uses only segment D000H. Single-window boards slightly degrade performance because of decreased memory. Single- and dual-window boards cannot be mixed in the same system. (On systems with an EGA, a dual-window board can be made to emulate a single-window board by removing a jumper at block E21-E22; anyone not having this jumper should contact Alloy directly for an upgrade).

The PC-Slave board controls its terminal's display by maintaining the monochrome video refresh buffer at segment B000H. The system checks the buffer several times a second and sends any changes back to the terminal via escape codes (code tables for various terminal protocols are maintained by the PC-PLUS NTNX software). If multiple changes are made between checks to the buffer, the system updates only

PC-PLUS FEATURES AND OPTIONS

PC-Slave/16N $895
8-MHz NEC V20 microprocessor
1MB memory
Math coprocessor support
Hercules graphics-compatible serial console interface
Serial interface
Installs in 8-bit slot

PC-Slave/286 $1,295
8-MHz Intel 80286 microprocessor
1MB memory
Math coprocessor support
Hercules graphics-compatible serial console interface
Serial interface
Installs in 8-bit or 16-bit slot

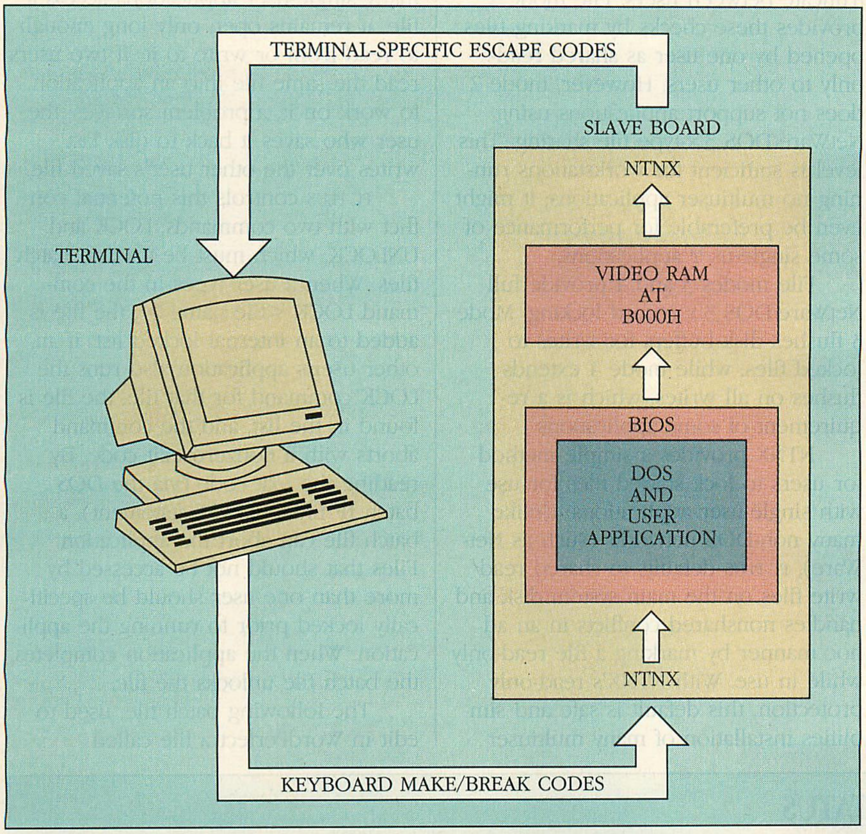
PC-PLUS Hardware Options
8-MHz 8087 kit for PC-Slave/16N $495
4-slot expansion unit $995
4-slot expansion unit for PS/2 $1,195
12-slot expansion unit $1,495
12-slot expansion unit for PS/2 $1,695

NTNX Network Software $495
PC-Slave/16N access to host peripherals
Disk caching
NetWare/DOS3.x file-sharing
Print spooling
User-to-user messages

NTNX/286 Network Software $695
Same features as NTNX, but works with PC-Slave/286

Optional Software
PC-PLUS Advanced Programmer's Kit $195
LINK-PC terminal emulation $195
MainLINE 3270 terminal emulation $2,995

FIGURE 1: ***PC-PLUS Slave-Terminal Communications***



NTNX receives terminal keystrokes and passes them to the application through port 60H, by generating software interrupts. It also monitors the video buffer at B000H and transmits screen changes to the terminal several times per second.

the current contents. Updates can sometimes result in temporarily garbled screen contents.

Monitoring of updates determines if they can be combined and sent more efficiently via special escape codes. A clear-screen operation, for example, requires only one code sequence. Because transmission rates are slow compared with buffer-checking time, monitoring can increase update speed.

The PC-Slave board handles user input by generating proprietary interrupt calls for keystroke make/break codes received from the terminal. PC-PLUS's NTNX software uses the proprietary interrupt to process keystrokes. After checking for keys used by the system, NTNX passes their messages through port 60H via software-generated interrupt 9H calls (see figure 1). The process exploits PC-Slave's proprietary nature, allowing network software to access keystrokes without generating interrupt 9H calls. This provides a stable and compatible environment for even the most sensitive software. This is important for programs that trap on interrupt 9H, such as Borland's SideKick and XyQuest's XyWrite III+.

SOFTWARE CONTROL

The NTNX software that makes the PC-PLUS system work runs from the host computer and uses DOS as its base (version 3.x is recommended for optimum file-sharing support). NTNX adds all resource-sharing functions to the operating system and provides several applications and utilities, such as print spooling, user-to-user messages, and diagnostics. It also provides a pop-up command menu (via Alt-F10) for accessing several network functions.

NTNX currently lacks electronic mail, but Alloy plans to offer an electronic mail package, called NXMail, in 1988. The software also lacks user accounts and directory-level security, although it supports simple drive-access security by allowing drives to be hidden from specified workstations. This, however, is more useful for protecting users from making mistakes than protecting sensitive data from unwanted access. Alloy's optional front-end password system, previously part of the Advanced User's Kit, is scheduled to be included in NTNX in 1988.

NTNX resides in the root directory of the system drive. It is loaded onto the host PC via both the NX.SYS device driver on the first line of the host's CONFIG.SYS file and the NTNX.EXE program, which normally is placed in the host's AUTOEXEC.BAT file. Once NTNX is loaded, the host boots a special NTNX-compatible version of COMMAND.COM (which is called COMMAND.SLV) on each PC-Slave board, using CONFIG and AUTOEXEC files ending in .U01 through .U31 (for users 1 through 31). Each user's CONFIG and AUTOEXEC files can be customized. NTNX comes with programs to duplicate default CONFIG and AUTOEXEC files for all users.

Because no user accounts exist and each user number is controlled by hardware, the easiest setup method is to configure all workstations the same, make a separate START.BAT file, and chain it to each AUTOEXEC file. This permits easy setup modifications and presents users with a familiar environment at all workstations. Customizations can be placed in individual batch files to be run from any workstation.

FILE SHARING

NTNX supports both Novell NetWare and DOS 3.x file sharing and record locking on shared drives by implementing the file-sharing interrupt 21H protocols (see table 1). A PC-PLUS network requires at least one shared hard disk. A second shared drive is permitted, but all other drives are accessible by only one user at a time.

In addition to the interrupt 21H calls, NTNX provides a set of PC-PLUS-specific function calls that use interrupt 67H for Orchid PC-Net-type file sharing and record locking, and interrupt 7FH for internal functions such as *datagrams* (internal packets of data that NTNX sends between users).

Alloy provides an Advanced Programmer's Kit that provides information on NTNX functions along with sample code. The *Advanced Programmer's Guide* (which is included in the kit) describes semaphores, NTNX datagrams, file locking, workstation control, disk-drive information, ATTACH/RELEASE resources, terminal control, and print spooler functions.

NTNX offers five levels of compatibility (file modes 0 through 4) that can be set with the FM.COM program; default is level 3, which is NetWare/DOS 3.x compatible. Protection increases at higher compatibility levels, but system performance decreases. Levels can be set individually at workstations to maximize their performance when full compatibility is not needed.

The lowest level affords no file or directory protection; none of the files can be locked except via Orchid PC-Net calls. All updates to common file allocation table (FAT) entries are written to disk directly, which could lead to disaster if two users simultaneously update the same portion of the FAT. Increases in file sizes from such updates are not available to other users who have opened the same file. Multiuser alternatives under level 0, therefore, are limited to PC-Net applications that preallocate files of a fixed size. In reality, level 0 should be used only if users always work in different subdirectories, which is essentially several single-user systems on a common disk drive.

File modes 1 and 2 allow two users to work in the same subdirectory, but not with the same files. Beginning with mode 1, Alloy provides *extended open support*, which supplies a copy of the FAT and the directory in the memory on each PC-Slave. Any changes that are made by one PC-Slave are immediately reflected on every other slave board's copy of the FAT, prior to the FAT's being written to disk.

File mode 1 does not provide checks for file write/delete/rename/truncate between users. File mode 2 provides these checks by marking files opened by one user as shared read-only to other users. However, mode 2 does not support applications using NetWare/DOS 3.x-type file sharing. This level is sufficient for workstations running no multiuser applications; it might even be preferable for performance of some single-user applications.

File modes 3 and 4 provide full NetWare/DOS 3.x record locking. Mode 3 flushes disk buffers for writes to locked files, while mode 4 extends flushes on all writes, which is a requirement of some applications.

NTNX provides a simple method for users to lock shared files for use with single-user applications. Unlike many non-DOS networks (such as NetWare), PC-PLUS defaults to shared read/write files on the main system disk and handles nonshared conflicts in an ad hoc manner by marking a file read-only while in use. With NTNX's read-only protection, this default is safe and simplifies installation of many multiuser software packages in which files must be shared read/write. However, when many single-user applications access a file, it remains open only long enough to read from or write to it. If two users read the same file into an application to work on it, a problem surfaces: the user who saves it back to disk last writes over the other user's saved file.

PC-PLUS controls this potential conflict with two commands, LOCK and UNLOCK, which must be used in batch files. When a user types in the command LOCK <file name>, the file is added to an internal locked list. If another user's application also runs the LOCK command for that file, the file is found in the list, and the command aborts with a nonzero exit code. By reading the exit code (via the DOS batch IF ERRORLEVEL statement), a batch file can abort the application. Files that should not be accessed by more than one user should be specifically locked prior to running the application. When the application completes, the batch file unlocks the file.

The following batch file, used to edit in WordPerfect a file called

UPGRADING PC-PLUS TO 286 STATUS

Alloy is staying in the performance race with its recent release of the PC-Slave/286, which upgrades the PC-PLUS network from an 8088-based system to 80286 class.

The 286 slave boards are functionally similar to their NEC V20-based PC-Slave/16N cousins. In addition to having an 8-MHz 286, each PC-Slave/286 is designed to fit into a 16-bit slot on the host's bus, essentially doubling the transfer rate for bus data. Both types of boards can be used in the same host, although in this case the 286 slaves must be placed in 8-bit slots for compatibility, thereby degrading their performance.

Installing the PC-Slave/286 is easier than installing the PC-Slave/16N, because fewer settings are necessary—only the base port address, user number, and single- or dual-window option need to be set. Cache memory set-up is handled by the new NTNX/286 software.

The only major change in the new software is in the main program, NTNX.EXE. All of the commands and utilities are the same as in the original NTNX. In addition, Alloy will be shipping an install program with NTNX/286 to simplify an already easy installation process.

For all the added processing speed that PC-Slave/286 offers, screen updates are still limited by serial communications rates, so no significant improvements are evident when using heavily interactive applications such as word processing. However, with its 16-bit data path for disk and cache transfers, the PC-Slave/286 really shines when it is used for processor- or disk-intensive applications, such as a multiuser database program.

Three 286-based slaves were put through the contention benchmark program used for this article. The new PC-Slave/286 boards are capable of using the power of a fast host computer; total system performance rose quickly as workstations were added to the test.

In addition, the 286 slave boards degraded the performance of the host more quickly than the 8088 boards did, to the extent that with two PC-Slave/286 boards plus the host, the host was slower than it had been in the 8088 tests. In fact, the performance of two 286 slave boards alone was better than the host plus one slave. This highlights the fact that the host must handle all disk processing in addition to its own application. As fast slaves are added, the host is forced to spend a larger percentage of its time servicing slave requests. In the PC-Slave/16N tests the host was still much faster than the slaves, even with three slave boards running, so this point was not as obvious.

As workstations were added, their effect on the performance of other PC-Slave/286 boards was similar to that noted with the 8088 slaves. When the host was included in the test, slave performance remained relatively constant as the host took most of the performance hits. With the host idle, however, the PC-Slave/286 boards lost some of their advantage, but they were able to maintain higher performance levels than were the PC-Slave/16N cards.

Overall, individual PC-Slave/286 performance improved 88-122 percent over PC-Slave/16N boards in similar configurations. It is important to note, however, that each of the 286 slaves were placed in 16-bit slots. The performance increases are less dramatic when 8-bit slots are used.

The cost of the 286 boards is $1,295, compared with $895 for the V20-equipped slaves). Thus, for $400 more, a PC-Slave/286 can deliver twice the performance.

—Gary Skiba

TEST.TXT, is an example of PC-PLUS's locking procedure:

```
WPL.BAT
     LOCK %1
     IF ERRORLEVEL 1 GOTO :ERR
     WP %1
     GOTO :DONE
     :ERR
     ECHO ... SORRY, FILE %1 IN USE
     :DONE
```

When the first user runs WPL TEST.TXT, the system adds TEST.TXT to its locked file list. When the next user attempts to issue the same command, the LOCK command returns an error, and the system performs an :ERR exit.

This approach has two problems: users must access common files via common batch programs, and they cannot use an application's file retrieval commands to load files because this would bypass the LOCK/UNLOCK procedure. The usefulness of LOCK/UNLOCK depends on the application; it is most effective for structured environments such as accounting applications.

SHARED RESOURCES

NTNX considers devices (CPUs, disk drives, tape drives, printers, modems), other than the main system disk drive on the host, to be shared read-only. It treats these devices the same whether they are centrally located on the host PC or connected to a PC-Slave. Write attempts to these read-only devices return a write-protect error.

Users can gain control of and write to these devices with the ATTACH command. A RELEASE command returns control of the device to the system. The ATTACH/RELEASE combination is the software equivalent of a hardware switch box for write access. It provides a powerful, safe method for sharing DOS resources. These commands also can be used to take control of another workstation (such as the host) by specifying the station number instead of the device name. This is useful for accessing programs that can be run only on the host.

For many batch operations, such as system backup, the entire ATTACH/RELEASE process can be placed in a batch file, making the resource access appear seamless. The ATTACH/RELEASE commands are also provided in the Alt-F10 pop-up command menu.

ATTACH/RELEASE is one of several options available through the SPOOL program for network printing and is especially helpful for printing on forms, when monitoring output is essential. Like most networks, printing is spooled to shared printers by redirecting BIOS interrupt 17H printer calls to send output to a file in a system directory (\SPLDIR in this case). To begin spooling output to the network printer, users run the SPOOL program from the command line (or a batch file).

If the printer is idle, users can choose the option of printing portions of a file immediately rather than waiting for the entire file to be available. If the printer is busy, the file will be spooled as usual. When the number of users is small, this direct printing option provides control similar to an attached printer, but without the attachment steps, waiting period, and chore of manually closing the spool file.

The SPOOL program allows for automatic closing of spooled output files by setting up a three-character auto-break sequence that, when encountered, automatically closes the spool file. With some modification of the word processor printer driver, the auto-break sequence can occur automatically at the end of documents. SPOOL also provides a time-out option, although the 10-second wait for files to enter the print queue can frustrate users, especially when printing several small reports from an application.

Missing from the NTNX print spool handler is an option for automatic break upon program termination to ensure that print jobs do not combine output from multiple applications. This could be accomplished by running the spool command from a batch file. The easier NTNX answer is to close the spool file via the Alt-F10 hot key.

Two types of local printers are available. Users can spool the second serial port on the PC-Slave board as device LOC1 or print a file to it using the SPRINT program. NTNX also prints to a local printer (LOC2) attached to an auxiliary port (separate from the main console port) on some terminals. (For example, the Link Technologies PCST/G graphics terminal has a parallel port with a 50KB buffer.) Users can print to the host and then to one local printer as soon as the host's spool file is closed, but they cannot access two local printers in this manner.

SENDING MESSAGES

PC-PLUS users can send messages to each other by running the SEND.COM program or by using Alt-F10. The syntax of the SEND command is:

SEND <user#> <message>

Like other networks, the message appears on the receiver's terminal when

TABLE 1: ***21H Function Calls***

ENVIRONMENT	
3DH	Extended open file
BBH	End of job status
D6H	End of job
DCH	Get station number
DDH	Set error mode
E7H	Return date/time
EAH	Return shell version
EEH	Get physical station number
FILE LOCKING	
C6H	Get or set lock mode
CAH	Log personal file (FCB)
CBH	Lock file set (FCB)
CCH	Unlock file (FCB)
CDH	Unlock file set (FCB/ASCIIZ)
CEH	Clear file (FCB)
CFH	Clear file set (FCB/ASCIIZ)
EBH	Log file (ASCIIZ string)
ECH	Unlock file (ASCIIZ string)
EDH	Clear file (ASCIIZ string)
LOGICAL RECORD LOCKING	
D0H	Log record
D1H	Lock record set
D2H	Unlock record
D3H	Unlock record set
D4H	Clear record
D5H	Clear record set
PHYSICAL RECORD LOCKING	
BCH	Log record (handle)
BDH	Unlock record (handle)
BEH	Clear record (handle)
BFH	Log record (FCB)
C0H	Unlock record (FCB)
C1H	Clear record (FCB)
C2H	Lock record set (FCB/handle)
C3H	Unlock record set (FCB/handle)
C4H	Clear record set (FCB/handle)
C5H	(A = 00H) Open a semaphore
C5H	(A = 01H) Examine a semaphore
C5H	(A = 02H) Wait semaphore
C5H	(A = 03H) Signal a semaphore
C5H	(A = 04H) Close a semaphore
DOS 3.x PHYSICAL LOCKS	
5CH	Lock or unlock file (handle)
MISCELLANEOUS	
DFH	Modify list device
E9H	Shell's get base status

Alloy's support software, NTNX, supports the interrupt 21H function calls that Novell NetWare and DOS 3.x provide for file sharing and record locking. Miscellaneous print and directory functions are also provided.

the sender executes the SEND command. Unlike many other networks, however, the receiver's application will not hang upon receiving the message because message functions bypass normal operations of the workstation (no interrupt 9H call is generated). When the receiver finishes reading the message and returns to the application, it remains intact. XyQuest's XyWrite III+, which hangs in similar situations on other networks, does not on PC-PLUS.

Sending messages using the hot key is simpler still: users press Alt-F10, select M for message, type in the receiver's user number, and then the message. The message appears on the receiver's terminal screen.

A powerful function of PC-PLUS is its ability to send keystrokes to other workstations, effectively providing background multitasking when other workstations are idle. This can allow a user to run a compile on another workstation while editing a different program (complete with a message when it is done); tell a workstation to release itself or its resources; or help a user in another room with a program by providing correct command syntax.

As with messages, commands can be sent from the command line (by using the syntax, CMD <user# > <keystrokes >) or by pressing the Alt-F10 hot key and selecting option C for command. The hot-key approach is best suited for starting applications or batch files remotely, while the CMD program is best for synchronizing workstations and ATTACH/RELEASE operations. During testing for this article, the CMD program was used to start each workstation from one batch file.

NETWORK SETUP

The system configuration tested for this review consisted of a 12-MHz Compaq Deskpro 286 acting as the host system, with a Compaq Enhanced Graphics Board and Color Monitor, a 40MB hard disk, and three PC-Slave boards connected to three PC-compatible terminals: the Link Technologies PCST/C ($649), Link's PCST/G graphics terminal ($749), and Kimtron's KT-70/PC. (PC-compatible terminals from Ampex, C. Itoh Electronics, and Wyse, as well as ASCII terminals from various manufacturers are also supported.)

These terminals are especially designed to work with PC-PLUS, as well as other PC-based multiuser systems, such as The Software Link's PC-MOS/386 and LAN-LINK. They support the entire character set and all monochrome video attributes inherent in the IBM PC. They also have AT-style keyboards. Because Alloy designed the PC-PLUS system to be used in addition to the host computer, this configuration effectively provides four workstations.

The serial port link between the PC-Slave and its terminal can be set at baud rates up to 38,400 bps, depending on the terminal and its distance from the host. Link's terminals support 38,400 bps, while Kimtron supports 19,200 bps. Alloy recommends using no more than 19,200 bps if the distance from the host exceeds 50 feet, because unamplified serial communications degrade after 50 feet.

Completing the PC-PLUS test configuration for this review were an Okidata 193 printer to evaluate the system's local and remote printing capabilities, and a Hayes 1200 Smartmodem to test communications capabilities.

> ***A powerful function of PC-PLUS is its ability to send keystrokes to other workstations, providing background multitasking.***

PC-PLUS is installed easily: first the hardware, then the software. Alloy's *NTNX Installation Guide* provides concise, step-by-step instructions on both procedures, although it lacks in-depth technical explanations of some settings, and the purposes of some options are initially unclear. Nonetheless, if the user retains the defaults, unless they are obviously inappropriate for a specific configuration, the documentation is a good guide through all the steps needed to bring the system up quickly.

To install the PC-Slave boards, users set the DIP switches on each board to reflect the board's base port address (which is used to pass control information to the board), unique user number, and memory configuration. The base port address must be the same for all boards installed in a system. Users generally should not change the 220H default unless they encounter installation problems.

In setting the memory configuration, users determine the amount of memory from the board that is to be used for DOS, a local RAM disk (when supported), and system cache memory. For testing purposes, all three slave boards were set to use 640KB for DOS, and the remaining 384KB for system cache. After the boards are installed in the host computer and connected to their respective terminals, the hardware installation is complete.

Software installation begins by copying three NTNX diskettes into the root directory of the host's hard disk. Alloy's DISKPREP program then performs a disk cluster optimization on the drives that are to be shared by the network, rearranging the files on the disk to be contiguous. Any copy-protected programs on the system's hard disk must be removed before running DISKPREP because it can move hidden files around and invalidate the protection scheme, rendering copy-protected software unusable. DISKPREP also creates a hidden DOS file on the disk that NTNX uses to manage the shared drive's FAT.

With the disk thus prepared, a call for device driver NX.SYS is placed at the beginning of the CONFIG.SYS file, and the operating system program NTNX.EXE is specified in the host's AUTOEXEC.BAT file. A batch file called DOSUTIL must be run for systems running under DOS 3.1 or 3.2. in order to install special NTNX versions of several DOS commands.

The last step before booting up the system is to run NXCNFG.EXE, the main setup program for NTNX. This executable file sets up and changes system parameters; including the number of slave boards recognized, the base port address on the boards, the default system drive and shared drives, cached disk drives, system cache parameters, lock buffers for multiuser applications, extended open support, and the maximum number of files that can be open at one time by all users.

NXCNFG also is used to specify each user configuration: the terminal type and communications parameters, hidden drives, 8087 support (an option on the PC-Slave), extended open support for the user, lock buffer size, cache/RAM-disk setup for the user's PC-Slave, DOS version, and automatic disk allocation for the user.

Automatic disk allocation is an option resulting from NTNX's unorthodox method of managing shared drives. Instead of providing all free disk space to all users, NTNX allocates free disk space clusters from the FAT equally among all users. Alloy claims that this is necessary under DOS to prevent cluster allocation problems. Therefore, with four users and 4MB of free space, each user can write only 1MB before

On April 2, IBM made PC history. Eight weeks later, we rewrote it.

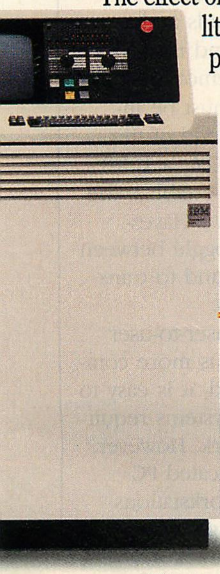
April 2, 1987.

To most computer add-on companies that was the date IBM dropped the bomb. With the introduction of PS/2 personal computers, the Micro Channel® revolution had begun.

The effect on our competitors? Many were prepared to do little more than count big inventories of emulation products for old PCs.

But we were ready.

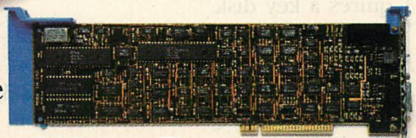
Within eight weeks we began shipping the first Micro Channel communications product. IDEAcomm 5251/MC.™ The board that connects the new PCs to the System 3X.



How did we do it?

The answer is simple.

We don't rely on outside development sources.



We have our own in-house R&D team. The same team that pioneered Surface Mount Technology (SMT) on computer cards. The very same manufacturing technique used heavily in the PS/2 design.

The result of these efforts is an add-on board that redefines the limits of PC communications. It provides 5292 Model 2 emulation. Support for 132 columns in 3180 emulation. And windows to monitor the real time status of multiple sessions on one PC screen. *Q38 Technical Journal* declared, ". . . while IDEA is delivering an emulation card that works completely, the other guys are just playing catch-up."

Which, historically speaking, is exactly the position we're inclined to take. For more information on the IDEAcomm 5251/MC, call 800-257-5027.

IDEAssociates™
The Leader in PC Technology.

IDEAssociates, Inc., 29 Dunham Road, Billerica, MA 01821, (617) 663-6878, Telex 4979780; France, Switzerland, Germany, United Kingdom, Hong Kong. IBM and Micro Channel are registered trademarks of International Business Machines Corporation. IDEAcomm 5251/MC is a trademark of IDEAssociates, Inc.

CIRCLE NO. 122 ON READER SERVICE CARD

receiving a disk-full error, a problem if disk space is limited. With automatic disk allocation enabled, however, a disk-full condition would initiate an attempt to take some of the host's free space and allocate it to the user until enough space (or no host space) is available, at which time a disk-full error would be returned. At that point, a program called DA can reallocate free clusters among users (or the host computer can be rebooted).

While inconvenient, this disk-allocation procedure prevents programs, such as the print spooler, from consuming all available disk space before a user can save the spreadsheet that the user has been working on for an extended period of time. If automatic disk allocation is disabled, the DA program must be run to reallocate free disk space among workstations as soon as the user's workstation space has been used up.

ALLOY COMPATIBILITY

Single-user, multiuser, communications, and graphics software were all tested for compatibility with the PC-PLUS network. Loading software onto the system requires simply installing it as usual from the host's A: drive, which can be accessed from a PC-Slave board because the A: drive is read only. If the software writes back to the A: drive, however, the user must ATTACH the A: drive to receive write privileges. This is a useful feature that is not available on the diskless workstations of most networks. It is especially important when using copy-protected software that requires a key disk.

Single-user tests. Single-user applications that performed as expected on the PC-PLUS system include Lotus 1-2-3, WordPerfect, XyQuest's XyWrite III+, Aker's MAGIC PC, and Ashton-Tate's dBASE III PLUS. The only difficulty encountered was an intermittent divide overflow error when attempting to run a program from the DOS prompt. Alloy is aware of the anomaly, but because the error never occurred while actually running a program, it should not pose a major problem.

Norton Utilities version 4.0 revealed one incompatibility. The NU program could run only on the host machine; when running on a PC-Slave board, the program aborted with an error in reading the FAT. An Alloy representative concurred that the problem was a result of the way NTNX allocates free clusters from the FAT equally among users. All of the other Norton Utilities programs worked fine, including Quick Un-Erase, in which a file can be successfully unerased only if the erased disk blocks have not been allocated to another user.

Several terminate-and-stay-resident programs were tested, including: Borland's SideKick and Turbo Lightning, Living Videotext's Ready!, Inset Systems' INSET, CED (a user-supported software command-line editor), and IBM's File Facility program, all of which worked with no problem. Used with the DOS SUBST command, the File Facility program allows the PC-PLUS user to emulate search path mapping; NTNX lacks any data file search path support such as NetWare's MAP command.

Multiuser tests. While single-user applications may be the bread and butter of many office needs, the real test of PC-PLUS's capabilities lies with multiuser software. Multiuser versions of dBASE III PLUS (LAN Pack) and MAGIC PC were tested; both support NetWare and DOS 3.x-type file sharing.

Alloy also offers a product, called MainLINE, that allows a PC-PLUS system to communicate with IBM mainframe computers.

The test involved first creating two delimited ASCII files with 1,000 records each; the files were loaded simultaneously from separate machines into an indexed database file using a DO . . . WHILE loop in dBASE, because the APPEND command works only on files opened for exclusive use. MAGIC PC (which supports multiuser applications by simply specifying that required files be record locked) was tested using the same delimited files. This exercise is a torture test of the multiuser capabilities of these database products in the PC-PLUS environment. Both products performed the load without difficulty.

Communications. PC-PLUS supports both traditional user-to-user communications and communications with a remote PC. Achieving successful user-to-user communications is as simple as connecting a modem to the COM2 port of one of the PC-Slave boards (or the host) and running communications software. During testing, DCA/Crosstalk Communications' Crosstalk XVI was up and running in five minutes. With PC-PLUS's ability to ATTACH to the workstation that has the modem, this setup is a powerful way to provide communications to all users (although it requires that the modem workstation be available at all times for this purpose).

Remote workstations can be linked in a number of ways. A remote PC-PLUS-compatible terminal can be connected to a modem, which makes it capable of communicating with another modem connected to a local PC-Slave board's console port. The local terminal user can then dial the PC-Slave's modem. Once the connection is made, the remote terminal operates as if it were physically connected to the local PC-Slave board (although at a much slower communications rate).

The simplest way to provide remote access to a PC-Slave board is to connect a modem to its COM2 port and run a remote workstation program (such as DCA/Crosstalk Communications' Remote, Meridian Technology's Carbon Copy PLUS, or Dynamic Microprocessor Associates' PCANYWHERE) on the PC-Slave. Alloy offers a tailored solution with its Link-PC product, provided that a PC-Slave board can be dedicated to remote access. Link-PC allows a user at a remote PC to work on the PC-Slave board in the local PC-PLUS system, with all commands executed on that board and all output displayed on the remote PC. Link-PC is especially useful for transferring files between the host and a remote PC.

Link-PC requires a special cable with a nine-pin male connector to plug a modem into the console port of the dedicated PC-Slave board. The NXCNFG program sets up the workstation's console port for the desired communications protocol. The workstation's AUTOEXEC file then must be modified so it can run a program called SLVLINK. This program loads and stays resident in the PC-Slave and is used to manage the link with a remote PC.

Another set of files must be installed on the remote PC end of the link. The main file is LINKPC, a 10KB resident program used to establish the link (either directly or via a Hayes-compatible modem), to toggle between local and remote modes, and to transfer files to or from PC-PLUS.

Unlike the standard user-to-user setup, Link-PC installation is more complicated, but once installed, it is easy to use. Link-PC is good for systems requiring substantial remote work. However, because it requires a dedicated PC-Slave board for remote workstations, the overall cost may be higher.

Alloy also offers a product that allows a PC-PLUS system to communicate with IBM mainframe computers. Called MainLINE, it allows PC-PLUS terminals to emulate IBM 3278 terminals communicating with the mainframe host via the PC host at speeds of up to 19.2-kilobits per second. MainLINE supports auto dial, auto answer, SDLC, and BSC communications, as well as 32 simultaneous mainframe sessions per PC-PLUS system.

Graphics. PC-PLUS does not shine very brightly in the area of graphics. While the network did support all of the graphics packages tested here (Lotus 1-2-3, Z-Soft's PC Paintbrush, and Inset Systems' INSET), performance leaves much to be desired. The problem is quite simple: at 19,200 bps, a 32KB Hercules-compatible screen can take 15 seconds to repaint. Highly interactive graphics-based products such as Microsoft Windows are not recommended for use on a PC-PLUS system.

A CONTENTIOUS NETWORK

PC-PLUS was tested to see how it performs under contention—that is, how quickly does a workstation's performance degrade as the load on the network increases? The test involved groups of workstations that began working simultaneously (synchronized via software) and for one minute repeatedly loaded a program that updates a single file (associated with that workstation). The number of repetitions performed in the one-minute period were tabulated. By loading a program for each update, the test exercises the PC-Slave's processor, as well as the system disk cache.

First, the host was tested by itself to produce a benchmark for comparison. Next, the test was run using one, two, and three workstations, both with and without the host. The results shown in figures 2 and 3 represent the number of iterations that were completed in one minute for each workstation, both in relation to the other workstations and as a reflection of the total load on the network.

The host Compaq Deskpro 286 easily outperformed the 8088-based slave workstations. As workstations were added, however, the host's performance degraded quickly. This was due primarily to the way PC-PLUS assigns lower priority to the host on system calls in order to keep it from monopolizing the hardware.

A more important measure, however, was the performance of the PC-Slave boards under the various loads (see figures 4 and 5). When the host was included, additional workstations tended to steal time away from the host, instead of from each other, so the addition of more users did not degrade slave workstations significantly. When the host was idle, individual workstation performance degraded more quickly, although the total amount of work performed by the system rose more quickly than the workstations' decline. Because the PC-Slave boards and their NEC V20 CPUs are only 8-bit processors, they were unable to make the most of the excess system capacity, because extra time was spent in loading the update program.

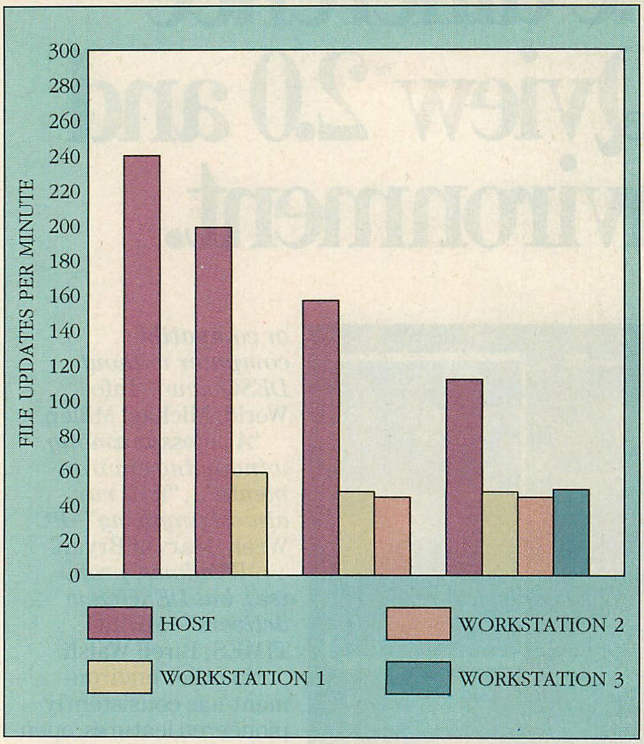
The benchmark results suggest that while PC-PLUS's bus architecture is well suited for providing fast access to network resources, its relatively slow processors (by today's standards) keep it from fully reaching its potential. In actual use, the first access to a program or file may take longer than expected, but subsequent accesses coming from the cache are quite rapid. This bodes well for performance from database programs, while that of casual word processing and other one-shot applications may prove disappointing.

ONE MORE SOLUTION

Alloy touts PC-PLUS as a high-performance alternative to networking PCs. It is not, however, a bargain-basement special. The price of one PC-Slave board plus a terminal is about $1,400. A four-workstation PC-PLUS system similar to the one tested here (considering the host to be one workstation that you already own) would cost $4,700. This is about the same price you would pay for a low-cost compatible-based LAN.

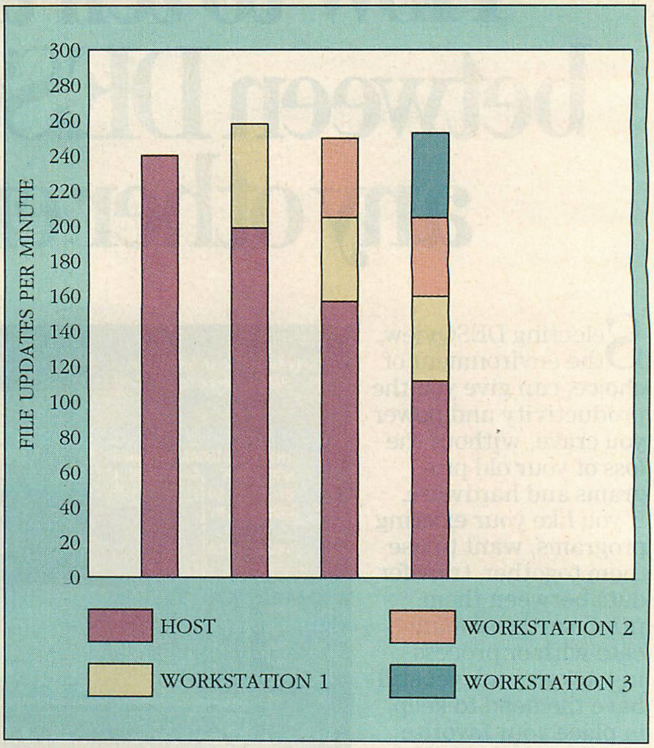
However, PC-PLUS is cost-effective in certain situations—especially in small work groups (2 or 3 users). In highly disk-intensive multiuser applications (such as dBASE III PLUS) where a network would get bogged down, PC-PLUS keeps running at a consistent pace. Because many office systems revolve around a few major applications such as accounting and database management, PC-PLUS's performance can make a big difference, especially when compared with a network of 8088-based machines. Because PC-PLUS is DOS based, its environment can be customized using readily available software. Further, PC-PLUS is a flexible system for turnkey applications, especially considering the ease with which workstations can share peripherals installed on the host. Idle workstations also can be controlled by other workstations to provide a simple form of multitasking.

FIGURE 2: *Host and Slave Performance*



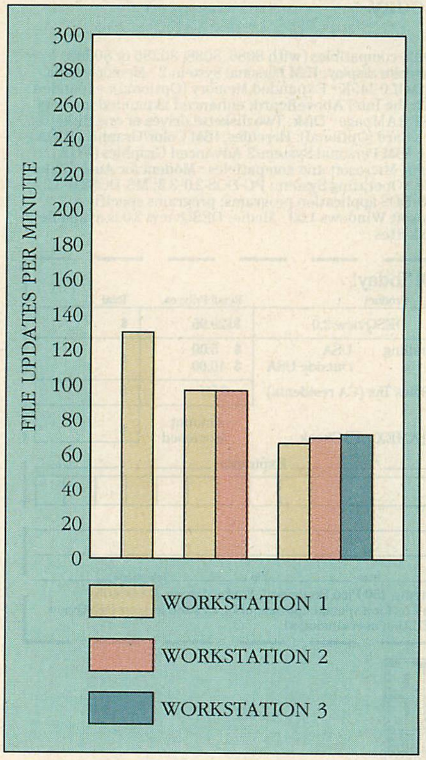
Decline in host performance is due to the lower access priority assigned to it as more slaves are active. When host is active, slaves steal time from it instead of each other.

FIGURE 3: *Total System Performance*



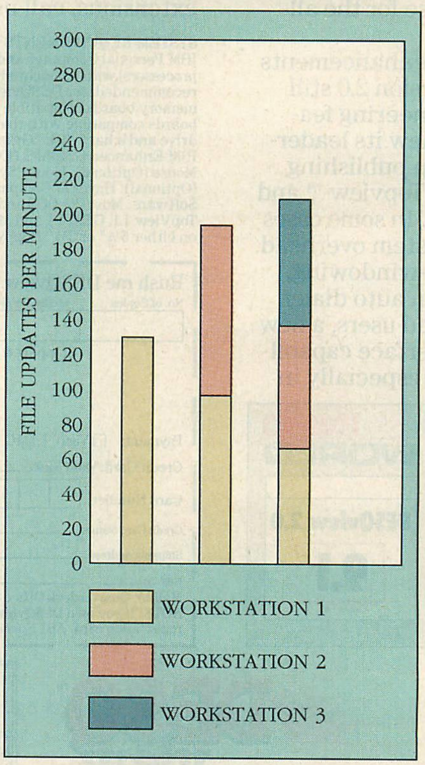
With host active, maximum system performance is achieved quickly. Additional processing performed as more slaves are active is offset by the decrease in host processing.

FIGURE 4: *Slaves Alone*



With host idle, performance of individual slaves degrades quickly as more become active, but the total work accomplished is still significant.

FIGURE 5: *Slaves Total*



With host's contention for resources eliminated, work performed by slaves does not equal total work performed with host working.

In what situations would PC-PLUS be less attractive? If highly interactive graphics are important, PC-PLUS's slow performance can be quite frustrating. Work groups that want to run Microsoft Windows as the primary operating environment should steer away from PC-PLUS. A traditional LAN might be more economical for an office that already owns several completely configured PCs. Finally, the lack of user accounts with directory access rights and electronic mail could make PC-PLUS inappropriate for widely distributed systems.

Networking schemes and products are in plentiful supply. Like all networking solutions, PC-PLUS has tradeoffs. It is appropriate for some situations; inappropriate for others. When the dust settles, however, Alloy's PC-PLUS is one more serious, effective, office solution from which to choose.

Alloy Computer Products, Inc.
100 Pennsylvania Avenue
Framingham, MA 01701-9990
617/875-6100
PC-PLUS
CIRCLE 339 ON READER SERVICE CARD

Gary Skiba is president of Skiba Data Systems, a consulting firm that specializes in multiuser systems and software.

No One Computer Can Meet All Of Your Needs, But One Company Can.

AST Premium/286

AST Premium/386

AST Premium Workstation

Priorities. They're different for every department. Some need more power in a computer that takes up less space. Others want state-of-the-art performance today with engineering innovation to keep up with tomorrow.

What's the answer? AST Research. We've been providing quality enhancement boards, peripherals and connectivity solutions for years. Now, with our complete family of high-performance computers, AST provides a single source for a broad range of needs. And model for model, AST Premium™ computers feature solid compatibility and superior performance for the best value on the market.

Take the AST Premium/386, for example. With 20 MHz operation, advanced architecture and AT® hardware compatibility, it's an incredible power tool for MS-DOS®, MS OS/2™ and XENIX® applications.

For general office computing, the AST Premium Workstation™ provides the power and flexibility of competing machines three time its size. And for extraordinary compatibility and 10 MHz, 0 wait-state speed, the acclaimed AST Premium/286 is at the top of its class, with a built-in upgrade path to the next generation of microprocessors.

For a more productive, versatile environment, make AST products your top priority. Need more information? Fill out the coupon below, or call AST Research, **(714) 863-0181**, ask for operator AA004.

AST® RESEARCH INC.

	AST Premium/386	AST Premium/286	AST Premium Workstation
Microprocessor	80386	80286	80286
Speed (MHz)	20†	10/8/6	10/6
Wait States	0-1	0	1
Standard Memory	Up to 2 MB	1 MB	1 MB
Expandable to	13 MB	13 MB	4 MB
Video Adapter	Optional	EGA/HGC (most models)	EEGA/EGA/HGC module
Expansion Slots	7*	7**	2
Fixed Disk	40/90/150 MB	20/40/70 MB	40 MB
Diskette Size	5¼" or 3½"	5¼" or 3½"	5¼" or 3½"

†Three software selectable speeds for timing-sensitive programs.

**One 32-bit dedicated to memory, three AT-compatible 16-bit multimaster and one 16-bit AT-compatible; and two 8-bit in standard models.*

***One 8-bit, six 8/16-bit including 2 FASTslots™; and four user slots in standard models.*

Please send me more information on the following:

- ☐ Premium/386
- ☐ Premium/286
- ☐ Premium Workstation
- ☐ Please have an AST representative call me.

Name ______________________

Title ______________________

Company ______________________

Address ______________________

City/State/Zip ______________________

Phone ______________________

AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992 Attn: M.C.

PTJ 2/88

CIRCLE NO. 199 ON READER SERVICE CARD

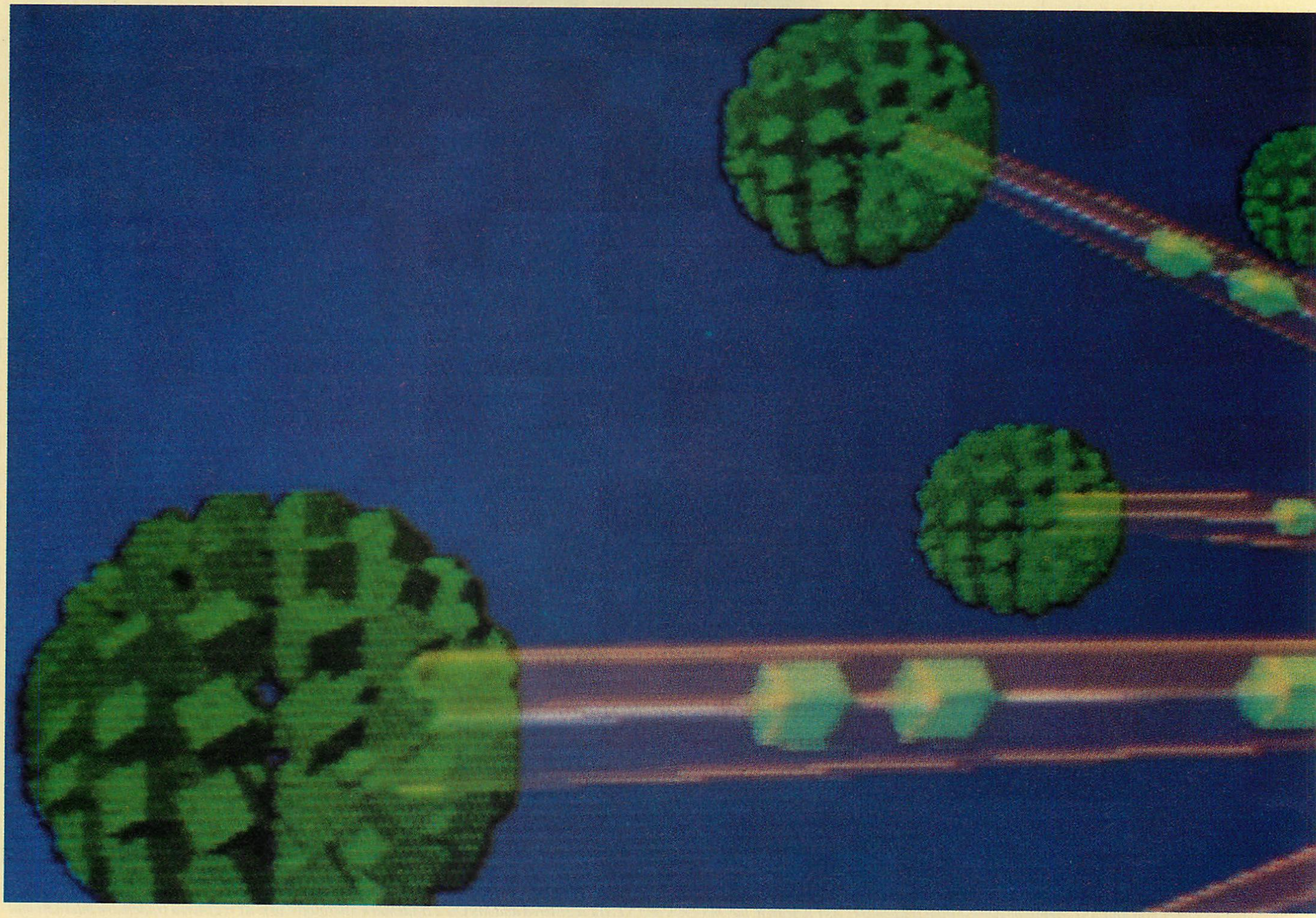
COVER SUITE:
SPEAKING SQL

DATA MANAGERS AS DEVELOPMENT TOOLS

The Multiuser Perspective

DAVE BROWNING

As the shared database becomes an integral part of the corporate environment, the multiuser features of data managers take on increased importance.



COMPUTER GRAPHICS • DOV JACOBSON

Multiuser demands being placed on data management systems in the PC environment are not static, nor are they simple. Requirements vary from business to business: Some demand highly sophisticated, flexible simultaneous access and updating, others depend on advanced security controls. Data managers now available for the PC provide varied multiuser features to match these needs.

Beginning in this issue, *PC Tech Journal* establishes basic criteria for measuring multiuser features of data managers. Multiuser evaluations will augment data manager reviews initiated in the August 1985 issue ("Evaluating Data Managers as Development Tools," Julie Anderson, p. 46), when multiuser features were primitive and rare, and shared databases uncommon. Future data manager reviews will evaluate multiuser features ("Earlier Data Manager Reviews," sidebar, p. 116).

Three data managers have been chosen for initial updates because they are representative of the varied multiuser features offered: Ashton-Tate's dBASE III PLUS 1.1, Borland International's Paradox 2.0, and DataEase International's DataEase LAN 1.02. Their general and single-user features were evaluated earlier (dBASE III PLUS 1.0 in "A Data Manager: The Evolving Standard," Dave Browning, May 1986, p. 166; Paradox 1.0 in "A Data Manager with Visual Queries," Will Fastie, April 1986, p. 154; and DataEase 2.5 in "A Data Manager for End-user Development," Dave Browning, September 1986, p. 146).

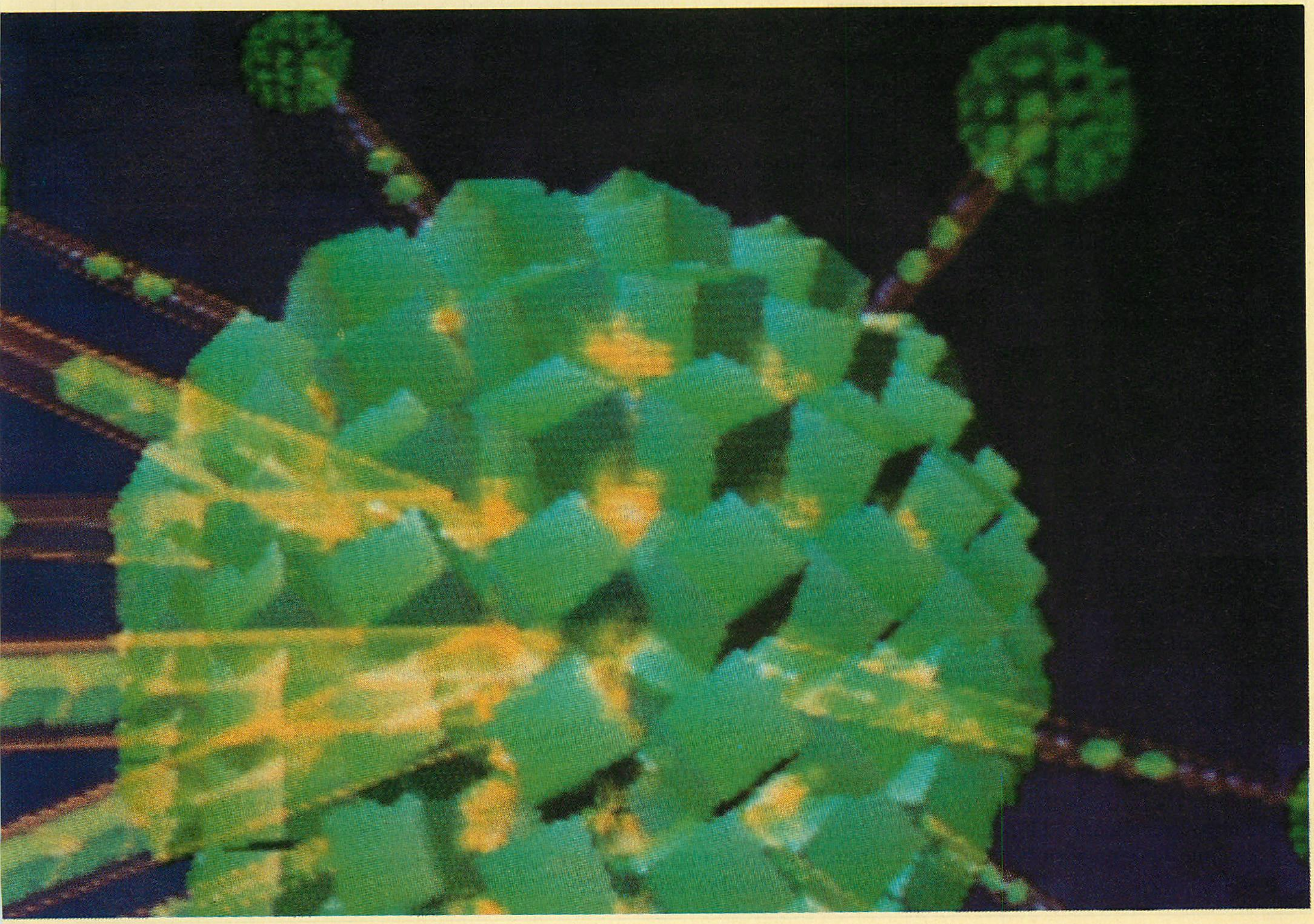
To provide updates on multiuser features while preserving integrity of existing reviews and allowing direct comparison between future and past reviews, minimum modifications to existing criteria and sample application specifications are made. Original sample application specifications are available on PCTECHline; criteria specific to multiuser operations are being added.

Multiuser features include data locking, transaction-processing, data integrity, security, and data backup. Data locking and transaction processing permit multiuser access to data simultaneously with minimal interference and confusion; security and backup features protect databases from accidental or unauthorized loss or corruption.

Numerous LAN configurations and potential conflicts with other LAN activities make it impossible to devise repeatable, accurate, meaningful multiuser performance benchmark specifications. In addition, data manager performance in single-user benchmark testing is not a reliable indicator of performance in a LAN environment. A data manager's interaction with a network operating system will affect efficiency of multiuser functions such as record and file locking and have a significant effect on multiuser performance (see "Data Managers and LANs," Dave Browning, May 1987, p. 54).

CRITERIA FOR GAUGING POWER

To demonstrate data manager data-locking capabilities, several operations have been added to *PC Tech Journal*'s sample application. A simple data entry



and update screen for maintenance of the Author file is developed, and two users simultaneously attempt to update the same record in the Author file. This is done in the data manager's interactive mode, if one exists.

Interactive locking capabilities. How does a user request that a record be locked? What indications are presented to a user when a desired record is locked by another? Is a user given the identity of the user (or station) that has a requested file or record locked? What happens if the desired record remains locked for an extended period of time? Is the user then aware that a locked record exists, or is the locked record hidden from view?

Does the data manager allow a user to enter changes to the screen before a record is obtained for update, then refuse to post the changes because another user already has modified the record? Is a user notified if a shared record being viewed is changed by another? Are all workstation screens refreshed to reflect updated data immediately? If not, how do users find out that a record has been modified since they last read it? Can multiple users read locked records simultaneously? Can users skip over or pass through locked records to access other data? Can records in a locked file be displayed by other users? Are some display modes inhibited?

Programmed locking capabilities. If the data manager has an application development programming mode, then the same multiuser update task is performed in that mode, under control of an application program designed for data entry to the Author table.

Among the questions to be answered are these: Does the data manager provide a rich variety of lock-request functions to accommodate different application requirements? How does the data manager notify the application of a successful lock request or an unsuccessful request? Will lock-request functions retry a lock attempt for a specified period of time or at a specified retry rate? Does the data manager provide an optional function allowing the programmer to notify the user of lock conflicts? Can a set of multiple records in a file be locked simultaneously, or must the entire table be locked? Does the data manager automatically manage the locking of associated files such as index files?

Reporting capabilities. The data manager's report writer is used to execute a report listing records from the Author file at the same time that an interactive or programmed update is done to at least one Author record. This demonstrates data manager performance under share-lock conditions.

Can reporting and editing be performed simultaneously on the same file? Does the report lock the whole file, a subset of the file, or each record individually as it is processed? Does the report stop and wait for a record to be unlocked? Does the report read and use the data from a record, even though a user has it locked? Does the data manager ignore the fact that a locked record exists, thereby presenting incomplete data in a report on a shared table?

Transaction processing. If the data manager provides transaction-processing features, transactions are created and executed simultaneously from two network stations using the following scenario. At one network station, a user discovers during entry of an article rec-

EARLIER DATA MANAGER REVIEWS

Since 1985, data manager reviews have focused on database design; data types supported; application development language capabilities; ease of changing field (column) length and adding and deleting fields from a file; data entry, update, integrity, security and protection features; and import/export facilities. The sample application tests reporting and querying capabilities and performance by using the following three files: Authors, Article, and Issues.

Data managers that were previously reviewed by *PC Tech Journal* and that are still on the market are listed below:

- Aker Corporation's MAGIC PC ("A Data Manager That Eliminates Programming," Victor E. Wright, October 1987, p. 122);
- Ansa Software's (now a subsidiary of Borland International) Paradox ("A Data Manager with Visual Queries," Will Fastie, April 1986, p. 154);
- Ashton-Tate's dBASE III PLUS ("A Data Manager: The Evolving Standard," Dave Browning, May 1986, p. 166);
- Business Tools, Inc.'s TAS-Plus ("A Data Manager for the Self-Reliant User," Jim Roberts, October 1986, p. 146);
- COSMOS's Revelation ("A Data Manager Designed for Complex Applications," Kent Phelps, February 1986, p. 160);
- Data Access Corporation's DataFlex ("A Data Manager for Diverse Environments," Chris Christian, August 1985, p. 52);
- Data Language Corporation's PROGRESS ("A Data Manager for Transaction Processing," Mark Karaman, July 1986, p. 142);
- IBM Corporation's Data Edition and Reports+ ("A Data Manager for Custom Reports," Dave Browning, January 1987, p. 150);
- Information Builder's PC/FOCUS ("A Data Manager with Mainframe Compatibility," William Casey, November 1985, p. 124);
- Metafile Information System's Metafile ("A Data Manager That Synthesizes Text and Data," V. Joseph Bowman, July 1986, p. 142);
- Micro Data Base System's KnowledgeMan/2 ("A Data Manager with Flexible Designs," Richard N. Aarons, June 1986, p. 156);
- Microrim's R:BASE 5000 ("A Data Manager with Kernel Code Generation," Steven Armbrust and Ted Forgeron, September 1985, p. 82);
- Oracle Corporation's ORACLE ("Managing Databases Mainframe-Style," Dave Browning and Hugo Blasdel, December 1987, p. 106);
- Q-N-E International's Q-PRO4 ("A Data Manager for Intelligent Screen Forms," Chris Christian, January 1986, p. 127);
- Relational Database Systems' INFORMIX-SQL ("A Data Manager with a UNIX Heritage," Angie Hansen, December 1985, p. 118);
- Relational Technology's INGRES ("Relational Power, PC Ease," Fabian Pascal, December 1987, p. 74);
- SoftCraft's Btrieve 4.03 and Xtrieve 3.0 ("A Data Manager with Language Flexibility," Burks A. Smith, October 1987, p. 104);
- Software Solution's (now DataEase International) DataEase ("A Data Manager for End-User Development," Dave Browning, September 1986, p. 146);
- UNIFY Corporation's UNIFY ("A Data Manager Strong on Administration," Jim Roberts, June 1987, p. 138);
- Zanthe Information's ZIM ("A Data Manager Using Entity-Relationships," Richard M. Foard, October 1985, p. 96).

—Dave Browning

ord to the Article file that the author name is missing from the Author file. Data entry to the Article file is suspended while the user adds the author name to the Author file. Meanwhile, a second station enters a second article by the same author.

This allows evaluation of the following transaction-processing capabilities: Does the data manager provide features to process transactions? Can these features be used in interactive mode by end users? What transaction processing functions are available in the data manager's programming language? Are duplicate author records added to the Author file? Does the second user detect the new Author record added by the first user before the first user's transaction is completed? If the first user backs out of the transaction (removing a temporarily added Author record), what happens to the second user's transaction? Can the second Article record end up in the database without the corresponding Author record?

Does the data manager detect deadlock if two or more transactions have locked some resources and are waiting to lock other resources already locked by another transaction? If the data manager detects deadlock, does it resolve the conflict and how?

If the data manager provides transaction processing, does it include provisions for transaction logging? Are before and after image journals maintained for crash-recovery purposes? If the data manager does not provide transaction processing, are any programming features, such as a function to lock a set of resources, provided to make it easier for a developer to program transaction processing?

Security. Data security is one of the most pressing problems in the shared database environment today. The criteria for evaluating multiuser data managers should include answers to these questions. What other multiuser data-integrity features are provided? What security features are implemented to protect data from access by unauthorized users? Can access control be provided at the record level? At the field level? What functions are provided for the network manager or database administrator to manage data security?

Development issues. Many important issues exist for development of multiuser applications. What LANs or other multiuser environments are supported by the data manager? If a data manager can be operated in single-user and multiuser modes, can multiuser applications be executed without change in single-user mode? Can multiuser applications, including locking and transaction-processing functions, be developed on single-user machines? Must multiuser functions be commented out in single-user environments?

Does the data manager provide a database server technique where the data management kernel program executes at a specified machine on the network? Can database tables be stored on multiple network shared volumes?

Can multiuser applications, including locking and transaction-processing functions, be developed on single-user machines?

Is data storage preallocated and, if so, how? Does the data manager support distributed database structures?

Many current data managers operate on middle ground offering some but not all multiuser features. For example, in dBASE III PLUS either one record or the entire file must be locked; in Paradox and DataEase LAN, specified sets also can be locked. Also, Paradox has automatic retry facilities for requesting locks and access, while the others do not. Of these products, dBASE III PLUS is most widely used, Paradox has highly sophisticated interactive multiuser features, and DataEase LAN employs a wealth of locking options.

AN OLD TIMER

dBASE III PLUS is a well-known and robust data manager based on the relational model. Among its many excellent features are an extremely flexible and powerful command language and an easy-to-use interface (assist) with pull-down and pop-up menus. Data are stored in tables and can be displayed in either row and column (field) or single-record format. This system's multiuser features take advantage of its display versatility by allowing editing in single-record or multirecord mode.

The data manager operates as either a single-user system, or a network configuration consisting of the administrator and one access program. dBASE ADMINISTRATOR and dBASE ACCESS programs, providing interactive features, replace the single-user dBASE program; copies of the access program are available to increase the number of simultaneous authorized users on the system. The LAN Pack increases the number of workstations by five.

Multiuser dBASE can be installed on many LANs, including Novell Advanced NetWare 1.0 or later, IBM Token-Ring or PC Network Program 1.0 or later, and AT&T StarLAN Network 1.0 and 2.0. In interactive mode, users can perform any operation in single-user dBASE.

Locking. Files can be opened for exclusive use (the default) or shared use by setting the internal dBASE exclusive flag to ON or OFF in the SET EXCLUSIVE ON/OFF command; exclusive use also is set by opening the file with the command USE EXCLUSIVE (see table 1). Exclusive use allows only one user to access a file and is required when one of the following commands is issued: INSERT [BLANK], MODIFY STRUCTURE, PACK, REINDEX, or ZAP. If a file has not been opened for exclusive use when one of these commands is issued, an error occurs. If one user has opened a file for exclusive use when another attempts to open the same file, the message "File is in use by another" is displayed. No indication is given as to which user has the file open.

Because EXCLUSIVE ON is the default, the user must execute the command SET EXCLUSIVE OFF when data sharing is desired. dBASE implements two levels of file locking for sharing data: automatic and explicit. dBASE III PLUS automatically attempts to lock a file when the user or the program executes commands working on the entire file: APPEND [BLANK], APPEND FROM, AVERAGE, BROWSE, COPY, COPY STRUCTURE, COUNT, DELETE ALL, INDEX, JOIN, RECALL ALL, REPLACE ALL, SORT, SUM, TOTAL, and UPDATE. If the file is in use by another user, an error message returns. For commands modifying database files, the automatic file-lock requirement during execution makes sense; for those that do not, such as COUNT and AVERAGE, it seems unnecessarily strict.

Explicit file locking is provided with the FLOCK( ) function from either programming or interactive mode. It prevents multiple users from simultaneously updating the same record and is commonly used when many records and indexes need updating in a shared file. When update is complete, the UNLOCK( ) command releases the file.

In addition to file locking, dBASE III PLUS permits explicit record locking during editing. Data can be modified in two modes: EDIT, in which the single record to be modified is displayed; and

BROWSE, in which many records are displayed and one chosen for update.

The EDIT command (or the equivalent CHANGE command) is the primary mechanism for multiuser record-by-record editing of a file opened for shared use. Records in shared files can be locked by the interactive user with the Ctrl-O key in EDIT mode; lock status is displayed on the screen status line of the workstation issuing the lock, and is updated when the user attempts an operation. In shared mode, the interactive user can read an unlocked record with the EDIT command, but cannot enter data to the screen before the record is locked—the system does not accept keystrokes and gives a warning beep if the user in fact attempts to enter them.

When a user editing a file with the EDIT command has a record locked, other users cannot land on, or pass through, the locked record. An attempt to use the PgDn or PgUp key to move to a record locked by another user yields the error message "Unable to skip. (press SPACE)." If a user attempts to use the GOTO [record#] command from the dBASE prompt to move to a record locked by another user, dBASE displays the error message "Record is in use by another," and a help prompt appears. When the user requests help, only the standard command syntax is provided. If the user then attempts to edit the record, dBASE III PLUS indicates that the user is positioned on the requested record, but presents a blank data-entry screen. Data cannot be entered into the blank screen, nor can the record be locked.

dBASE III PLUS locks the entire file when the user enters BROWSE mode, which allows editing in table format. Because the user can access only one record at a time (by moving the cursor to the record's row on the screen), dBASE's automatic locking of the entire file, rather than each record as accessed, is too restrictive.

If a user attempts to edit data or execute a report on a file that another user is viewing with the BROWSE command, an error message indicates that the file is not available. Even so, others can view a record using the EDIT command and view or print records with the LIST and DISPLAY commands. LIST and DISPLAY provide limited formatting capability, wrap at the right-hand screen boundary when data lines exceed 80 characters, and do not allow the user to scan in reverse. If a user is editing records using the EDIT command but has not yet locked a record, another user can enter BROWSE mode. Since BROWSE locks the file, the user cannot lock a record in EDIT mode.

Locking functions operate in the same fashion, whether they are initiated from the console in interactive mode or included in a program executed by the dBASE III PLUS command interpreter. dBASE views all failed lock attempts as errors and generates an error number and message. In interac-

TABLE 1: *dBASE Locks*

LOCK	COMMAND/ KEY SEQUENCE
EXCLUSIVE	SET EXCLUSIVE ON[a]
	USE EXCLUSIVE
SHARED	SET EXCLUSIVE OFF
File	FLOCK
Record	RLOCK, LOCK
	Ctrl-O (with EDIT)

[a] *Program default.*

All files default to exclusive use, unless the SET EXCLUSIVE OFF command is issued for shared use. In shared mode, file and record locks are automatically locked for some commands or explicitly locked using Ctrl-O or lock commands.

tive mode, the error message is displayed to the user; in programs, the error condition can be trapped by using the ON ERROR command setting to trigger a programmer-provided error-handling routine. The error number and message are available to the program through the use of the ERROR( ) and MESSAGE( ) functions.

dBASE III PLUS does not retry failed lock requests automatically or notify users or programs when a previously unavailable resource is released; the user or program must reissue access requests. Files such as index or memo files associated with a data file are automatically locked by dBASE when the data file is locked by the user; there is no conflict or contention for these files because users cannot explicitly request that dBASE lock them.

In a multiuser environment, dBASE III PLUS requires special efforts to protect data from being changed when they are being written to a report. When a report developed for the dBASE III PLUS report writer is executed against a file, dBASE attempts to lock the entire file, even though only a subset of the file might be specified by conditions in the REPORT command. Reporting and editing are mutually exclusive. If a user is editing the file with a record locked, a report using that file cannot be started; when the report is running, the file cannot be edited (but can be viewed in EDIT mode).

dBASE III PLUS permits data and associated files to be opened simultaneously in work areas. The entire file or only one record in each open file can be locked at a time, either in interactive mode (while in EDIT) or under program control (using the RLOCK( ) function). Records cannot be locked in sets. It is not possible, for example, to lock a set of records in a file associated with a particular customer while a data change is applied; the entire file must be locked. No programming technique works around this restriction, which is especially inconvenient when several users need simultaneous access to a data file, with each operator working with a different customer.

The UNLOCK command removes any lock associated with the currently selected work area; UNLOCK ALL removes locks in all work areas. Closing a file or ending the dBASE program also removes all existing locks. If dBASE aborts at a user station with a record or file locked, other user stations must close the locked file and reopen it to clear the lock status. They are not alerted that a workstation is down, but only receive a message that the requested record or file is locked.

When developing multiuser applications, programmers must provide code to manage locking functions. If a multiuser program is executed on a single-user system, all requests for locking and exclusive use evoke a response indicating that the function has executed successfully. Developers can develop programs on single-user systems, but must test them on multiuser systems. The LOCK( ), RLOCK( ), and FLOCK( ) functions all return a true for successful access or a false for failed access. The programmer can place the function in a loop and repeatedly attempt the lock until successful, until a specified number of failed attempts occurs, or until a specified amount of time elapses. dBASE does not provide automatic retry at a specified time interval or rate, nor does it allow a delayed retry to be set to reduce lock-request traffic on the network. Typical dBASE code to lock a record, similar to that suggested in the dBASE manual, is given in the following:

```
GOTO editrec
t = 1
DO WHILE .NOT. RLOCK() .AND. t < 250
t = t + 1
ENDDO
IF t = 250 .AND. .NOT. RLOCK()
DO [error routine program]
ELSE
DO [record maintenance program]
ENDIF
```

This approach is not ideal because the network is flooded with lock requests as fast as dBASE can execute the DO WHILE loop. (Ashton-Tate should add time delay within the loop in a future release.) In addition, an error occurs when the program attempts to position to the desired record. The ON ERROR setting can specify an error-handling routine executed when an attempt is made to position the file to a locked record or when another error is encountered. RETRY executed from the error routine reattempts the command that causes the error, while RETURN executes the next command. The same approach works for interactive editing commands, such as EDIT and BROWSE.

dBASE's overall locking scheme is complex, restrictive, and confusing. For example, exclusive use is required in some situations where a less-restrictive explicit file lock is adequate. Although exclusive mode is desired for operations that change a file's format (which should not be accomplished if more than one user has the file open), other operations that do not affect file structure, should require only an explicit file lock with the FLOCK( ) command. dBASE also requires an exclusive lock to refresh an index file using the REINDEX command, but not to create or overwrite one.

BROWSE can be restrictive because it locks the entire file even though the user might be working on only one record. Shared editing of a file cannot be accomplished while any user is in BROWSE, nor can reports be written. In EDIT mode, shared editing can occur if individual records are locked, but locked records cannot be passed by and reports cannot start until an explicit file lock is placed. Under dBASE, implementing an effective locking strategy in programmed applications requires detailed programming.

Deadlock. dBASE III PLUS does not detect nor recover from deadlock. The data manager relies solely on the programmer to avoid such situations.

Transaction processing. This product provides no transaction-processing capabilities. Application programmers

must define transactions, provide desired transaction log and journal recording, and make provisions for transaction backout and recovery. Transaction-processing features can be programmed using the dBASE locking functions, but must be programmed individually. Transactions involving more than one record in a file require locking the entire file because dBASE does not allow one program or user to lock a set of records in the file.

Security. dBASE III PLUS provides a security program named PROTECT. It controls login access to the dBASE program, access to files and fields through assignment of privileges and user levels, and encryption of data. File-level access control assigns combinations of extend, delete, read, and change privileges to user levels. Field-access choices include full, read-only, and none, for each user level. Access controls are assigned to database files only; other file types, including .DBT memo field data storage files, are not encrypted. Index files associated with a data file are assigned the same privileges as the data file.

When the PROTECT program is used to create a file privilege scheme for a data file, the data file is automatically encrypted. PROTECT encrypts from a source file to an encrypted copy with a .CRP extension; the manager is warned to remove the nonencrypted data file copy and rename the encrypted file with a .DBF extension.

MODERN AND INTELLIGENT

Paradox 1.0 was introduced as a single-user data manager, based on the relational model. It boasts a Lotus 1-2-3-style menu interface and a Query-By-Example (QBE) facility. Paradox 2.0 retains all features of its predecessor, plus sophisticated multiuser features.

Paradox 2.0 comes in a single-user version or a network version with one authorized simultaneous access. Use of additional Paradox packages increases the number of simultaneous users. Borland's Network Pack adds six workstations. Paradox runs on many LANs, including Novell Advanced NetWare 2.0A or later, IBM Token-Ring or PC Network with IBM PC Local Area Network Program 2.1 or later, and AT&T StarLAN Network 1.1 or later.

Data are stored in tables that are viewed one record at a time in FORM VIEW or multiple records at a time in TABLE VIEW. Each table can have objects associated with it, such as forms, report formats, validity checks, indexes, and image settings. A table and its associated objects are called a family.

Locking. Paradox provides a rich set of locking features, oriented toward maximizing access to shared data, for use in both interactive and programming modes. It has four basic types of file locks: Full Lock, Write Lock, Prevent Full Lock, and Prevent Write Lock (see table 2). Its intelligent file lock-management system defaults to the least-restrictive lock needed to protect data during a specific operation, but users with a full understanding of all four can customize data sharing based on specific needs.

Full Lock prohibits access to an object by other users for any purpose. Write Lock prevents other users from changing an object in any way, although they can still use it. Prevent Full Lock prevents others from setting a Full Lock on an object; Write Locks can still be placed on the object. Prevent Write Lock prevents others from starting an operation that requires either a Write Lock or a Full Lock on the object.

Full Lock is most restrictive, while Prevent Full Lock is least restrictive in that it allows any operation except those requiring Full Lock. Paradox also provides family locks to lock all objects in a family and record locks to lock individual records. Users can place multiple locks on a table simultaneously from a single workstation or from multiple workstations.

Paradox automatically applies record and file locks in interactive mode, but users can override the defaults. Paradox automatically applies a Prevent Full Lock, for example, if a user selects the Insert (record) or Delete (record) option from the menu. As soon as the user selects DO-IT!, the Full Lock is set automatically until the action on the object is complete. Up to that time, other users accessing that file can perform operations not requiring a Full Lock. Another user, totaling or averaging numeric data, might want to ignore Paradox defaults and explicitly set a Prevent Write Lock to block others from changing a value in the course of his calculations.

Record locking also permits multiple users to edit tables simultaneously in Coedit mode, in which multiple users can read records in either table or form view. Paradox locks a record only when a user attempts to change it. If another user has the record locked, the data manager displays a record-locked message and the login name of the user who has it locked. In addition, users can explicitly lock and unlock records by using the lock toggle key sequence, Alt-L.

TABLE 2: ***Paradox Locks***

LOCK[a]	VIA MENU/ KEY SEQUENCE	VIA COMMAND[b]
Full	Tools/Net/Lock/FullLock	LOCK *table* FL
Write	Tools/Net/Lock/WriteLock	LOCK *table* WL
Prevent Write	Tools/Net/PreventLock/WriteLock	LOCK *table* PWR
Prevent Full	Tools/Net/PreventLock/FullLock	LOCK *table* PFL
Family	Tools/Copy/Table Tools/Copy/JustFamily	
Record	Alt-L	LOCKRECORD

[a] *Default is different for each operation as detailed in Paradox* User's Guide.
[b] *Table means specific table name.*

Paradox automatically adjusts locks on objects for each operation; in addition, locks can be explicitly placed using menu selections or PAL commands.

In coedit, users can read a record that another has locked for modification. Paradox provides a refresh feature to update screens of all users sharing data as one user changes them. Three types of refresh modes are available: explicit refresh, automatic refresh at fixed intervals, and automatic refresh as needed. A user can cause an explicit refresh of data on all screens at any time with the Alt-R key. Paradox's automatic refresh can be set at time intervals from one second to one hour. It always refreshes a record on a user's screen as soon as the user attempts to modify the record even between preset refresh intervals.

Reports can be executed on a table being changed in the Coedit mode; Paradox extracts a "snapshot" of the data required for the report and then executes the report. If a user changes data required for the report while the snapshot is being taken, Paradox displays a message to this effect and restarts the snapshot. For reports against volatile tables where many changes are being made, the report snapshot might restart many times between data changes. Paradox cannot be forced to take a snapshot while changes are being made. A user can explicitly Write Lock a data table before executing the report to prevent data changes during the snapshot.

Applying explicit Full Locks and Write Locks can improve processing performance when simultaneous data access or update is not required. Aware that updating is not possible, Paradox buffers portions of the file and does not check repeatedly for changed data.

In programming Paradox applications in the Paradox Programming Application Language (PAL), the LOCK and UNLOCK commands set and clear locks. Command parameters can be set to specify tables to be locked and lock types. A Paradox variable, **retval**, returns true if a lock attempt is successful, false if unsuccessful. If an attempt fails, the ERRORCODE, ERRORMESSAGE, and ERRORUSER functions determine what happened, why, and who caused it. The UNLOCK [ALL] command cancels selected or all explicit locks in effect; the RESET command also releases all explicit table locks, and it should be used at the end of an application program.

As with many procedural languages, successful lock management programming requires loops and error checking; the Paradox *PAL User's Guide* suggests loops as follows:

```
WHILE (True)
LOCK "Employee"
IF (retval)    ; successful?
THEN
QUITLOOP    ; continue code beyond loop
ELSE
MESSAGE ERRORMESSAGE() ; show
                       ; message
  . . . any other error processing . . .
ENDIF
  . . . any other processing for unsuccessful
  lock
ENDWHILE
  . . . successful lock processing section
```

Programmers can establish sleep periods within the loop to avoid flooding the network with lock requests. The SETRETRYPERIOD command can be used to direct Paradox automatically to continue attempts to lock for a specified maximum time. No corresponding command exists to set a retry rate.

Paradox successfully provides a flexible and automatic data locking strategy for data sharing on LANs in interactive mode. In most cases users do not need to learn the intricacies of lock types; shared data are presented with as few restrictions as possible; and users are notified of the currency of data and, when they cannot be accessed, who has them locked. Locking features for applications programming are good; set locking and the ability to establish multiple locks on an object are highly valuable. Borland might wish to add a function to set the lock retry rate in conjunction with the SETRETRYPERIOD to minimize loop coding for lock attempts.

Deadlock. Paradox does not detect nor recover from deadlock situations. However, the Paradox LOCK command permits programmers to specify a set of resources to be locked; Paradox either locks the full set or nothing. Careful use of set-locking commands can avoid situations that lead to deadlock.

Transaction processing. Paradox does not provide transaction processing in the normal use of the term, and transaction commit and rollback are not provided. However, it does provide an interactive data-entry mode whereby data records are entered to a temporary file and posted to the main file when the user presses DO-IT!. Data entry can be aborted prior to posting if desired. Posting detects duplicate record conflicts in keyed files and presents the offending records to the user for resolution. This data-entry mode provides rudimentary transaction processing for data entry only.

Transaction-processing features can be programmed individually using the locking functions provided by PAL. Paradox's ability to lock a set of multiple records in a single file can be useful in programming transaction processes, as can the ability to request, with a single command, that Paradox lock a list of resources such as records and files. This set-lock feature can limit the possibility of deadlock.

Security. Security is provided by the Paradox Protection Generator, which can establish table access using passwords for individual users. Each user has a password, as does each protected table. The Protection Generator manages a password database that links each user to all authorized tables; users need only to know their individual passwords, not table passwords. This mechanism simplifies security measures because table passwords do not need to be changed when employees leave a company. Each user can be granted individually tailored sets of rights to tables via the TablePasswords Form. Table rights include: All, Entry (add data), InsDel (add or delete), Update (non-key fields only), and ReadOnly. Field rights (All, ReadOnly, None) can be granted for individual fields. Rights can be extended to associated objects.

Each user can have a login script, which Paradox automatically executes when the specified user logs in to Paradox. Passworded tables and scripts are encrypted. Encrypted scripts require the entry of the password to read or modify the script, but they can be executed without specifying the password.

Locking features for applications programming are good; set locking and establishing multiple locks on an object are valuable.

VARIED LOCKING STRATEGIES

DataEase LAN 1.02 is based on a combination of the relational and entity relationship models of data management in which forms (data or files) are linked via user-defined relationships between fields having like values. It is distinguished by its use of forms, visual representations of data that define database structures, data-entry screens, fields, relationships, and so on. A DataEase program, which is the query section of a full report, is written in DataEase's nonprocedural development language. Multiuser access is managed with a complex locking strategy.

DataEase LAN 1.02 runs on Novell Advanced NetWare 86/286 2.0A local area network. It can be installed on a PC as a single-user program or on a LAN as a multiuser system with one authorized user. Workstations can be added by purchasing additional copies of DataEase LAN or a Workstation Upgrade that adds three workstations with accompanying documentation. Workstations added with DataEase LAN can create local databases and access both local and shared databases; those added with the Workstation Upgrade can access only shared databases.

Locking. DataEase LAN provides a wealth, perhaps even an overabundance, of locking options. A user specifies a locking strategy that applies to all databases in an installation using the Multiuser Locking Options Configuration Screen available from the system configuration menu. A user assigns a locking rule (Shared, Exclusive, None, Lock Records, Lock Files, Lock Nothing) to each of the five operations (form definition, record entry, loading/saving reports, report data processing, and record selection).

Because locking rules are stored in system files, they apply to all databases (up to 26) defined within the system. A problem can occur if parameters are changed while more than one user is active in the database; new parameters are immediately applicable to the user that set them, but other users still operate under old parameters until they exit and reenter DataEase LAN. Table 3 shows available locks and DataEase LAN defaults. Entire databases can be locked under some circumstances, such as backup/restore or application installation.

Form definitions can be locked for Exclusive or Shared use. DataEase LAN never allows a user to modify a form definition while another user is viewing it, regardless of whether the lock is Shared or Exclusive. If attempted, a Lock Timeout error occurs. The same locking strategy applies to loading as well as saving reports.

For record entry, locking rules can be None, Shared, or Exclusive. When None is specified, multiple users simultaneously can view, modify, or delete a record; if one user modifies it, others who have read it will not be aware of the modification unless they try to change the record. In this case, the error message "Record modified since it was read" appears. Shared record locking restricts users from modifying or deleting a record being viewed by another user. The Lock Timeout message appears until such time that the second user moves away from the record. Exclusive record locking prevents multiple users from viewing the same record simultaneously; the Lock Timeout message appears instead of the requested record.

For report processing, including programming, rules are assigned for report data locking and record selection locking. The sharing of form data accessed by reports referencing the form is controlled by the rule chosen for report data locking. Lock Records prevents modification of records selected by the report; Lock Files prevents modification of all records in a file, whether or not selected by the report. If the Lock Records rule is se-

lected, DataEase LAN locks records one-by-one as the report program processes them using the locking rule (None, Shared, or Exclusive) selected earlier for record entry. If Lock Files is chosen, all files referenced in the report program are locked. Queries reading records permit other users to read, but not update, records in a file; queries adding, modifying, or deleting records block access by other users.

Rules assigned for record selection lock the data when DataEase LAN is matching the records in the file to selection criteria specified in the report. This controls data modification between the time the report selects data and processes them.

If the report data locking rule is set to Lock Files, then all files from which records are selected are locked. If the report data locking rule is set to Lock Records, then Lock Nothing, Lock Records, or Lock Files can be selected.

Lock Nothing allows other users to modify or delete records that the report has selected but not yet processed. Lock Records prevents modification or deletion of records selected by the report; Lock Files prevents modification or deletion of all records in a file, whether or not they were selected. Records may be viewed during report processing in all cases.

Locking options set through the configuration screen apply to interactive operations and programs written in DataEase's nonprocedural Data Query Language. When a lock attempt is unsuccessful for five seconds, DataEase LAN sounds a beep and displays a window message such as:

```
Lock Timeout:
Waiting for <record, form, report, etc. >
Retry attempt number <n > , Press F4
  EXIT to abort
```

This message is displayed until the user releases the locked item so the operation can proceed, or until the user aborts the process. (Some processes cannot be aborted and DataEase LAN will not recognize **Press F4** in such cases.) No indication is given as to the identity of the user or process that has locked the requested resource.

DataEase LAN documentation suggests that the record selection locking rule be set to Lock Nothing or Lock Files when sets of selected records are large. Lock Files maintains data integrity and prevents using up all Novell locks, but restricts users from modifying data while reports are executing. Lock Nothing provides maximum simultaneous access but opens the possibility of data integrity corruption if users modify records at inappropriate times during report processing.

TABLE 3: *DataEase Locks*

LOCK	LOCKING RULE
Form Definition	Shared Exclusive[a]
Record Entry	None[a] Shared Exclusive
Report Definition	Shared Exclusive[a]
Report Data[b]	Lock Records[a] Lock Files
Record Selection[b,c]	Lock Nothing Lock Records Lock Files[a]

[a]Program default.
[b]Lock used is Shared for "List records" reports, or Exclusive, if report adds, modifies, or deletes records.
[c]Applicable only if Report Data's locking rule is Lock Records.

DataEase has an extensive locking strategy that is set up for the implementation from the Multiuser Locking Options Configurations Screen. The locking rules can be temporarily overridden by embedded programming language statements.

Programmers can use three language statements (lock, unlock, and query selection) to override temporarily the default locking rules in the programming language embedded in DataEase full reports. Lock can be used to lock all files, individual files, or a selected record. It prohibits access by other users to files referenced in the query during add, modify, or delete operations, but allows viewing during List Records operations. DataEase LAN documentation shows a query that lists each sales department employee's salary as a percentage of the department's total salary. The set of employee records for the sales department is selected twice, once for totaling salaries and again for listing records. The file must be locked as shown here:

```
Lock All Files.
define "Total" number.
Assign temp Total : = sum of Employees
  with
(Department = "Sales") Salary.
for Employees with Department = "Sales";
list records Salary / temp Total * 100.
end
list records temp Total.
```

This prevents records from changing between selections. In order to lock individual files, however, DataEase LAN provides the following sample code:

```
Unlock All Files.
. . .
Lock File Employees Shared.
define "Total" number.
Assign temp Total : = sum of Employees
  with
(Department = "Sales") Salary.
for Employees with Department = "Sales";
list records Salary / temp Total * 100.
end
Unlock File Employees.
. . .
list records temp Total.
. . .
```

A selected record is locked from the time it is selected to the time the next record is selected. Programmers can, however, use the unlock record statement to release a locked record before the next selection. This frees the initial record for access by other users while the program performs processing steps not involving that record. This option applies only if Lock Record is in effect for report data locking.

The unlock statement can be used to override the configuration setting for the report data locking operation and unlock all files for a program, a single file within a program, or the selected record. Query selection overrides the record-selection locking rule and allows it to be reset to Lock Nothing, Lock Records, or Lock Files for an individual report program.

DataEase LAN provides no programming commands or functions to test lock success or failure; a program simply waits for lock success or for the user to abort the program.

Locking strategies possible with DataEase LAN are flexible and varied. The data manager tries to use record locking as much as possible to permit maximum data access by all users, but use of record locking over file locking in report processing is inherently risky. Locks are managed by the network operating system; the number of simultaneous locks available from a Novell network file server are limited. Query programs operating in record lock mode can select too many records for the server to handle, which causes the query to hang.

Deadlock. DataEase LAN does not address the issue of deadlock.

Transaction processing. Because locks can be specified for *sets* of records or files and remain in effect until the data processing has reached completion, some transaction processing is implicit in the DataEase programming language.

TABLE 4: ***Multiuser Capabilities***

	ASHTON-TATE	BORLAND	DATAEASE
PRODUCT	dBASE III PLUS	Paradox	DataEase LAN
VERSION	1.1	2.0	1.02
PRICE	$695	$725	$700
Identify station locking file/record	○	●	○
Lock set of records	○	●	●
Permit simultaneous editing, reporting	○	●	●
MULTIRECORD EDIT			
Lock record without locking entire file	○	●	N/A
Allow others to view locked record	●[a]	●	N/A
Scroll by locked record	N/A	●	N/A
GO TO locked record	N/A	●	N/A
Automatic refresh of all screens	○	●	N/A
SINGLE RECORD EDIT			
Share file	●	●	●
Toggle record lock	●	●	●
Allow others to view locked record	○	●	●[b]
Pass by locked record	○	●	●[b]
GO TO locked record	○	●	●[b]
TRANSACTION PROCESSING			
Commit and rollback	○	○	○
Before/after image journal	○	○	○
DEADLOCK			
Detection	○	○	○
Recovery	○	○	○
SECURITY			
Data encryption	●	●	○
User login	●	●	●
Field level access control	●	●	●
Program access control	●	●	●

● = Yes ○ = No N/A = Not applicable

[a] When in EDIT mode.
[b] Only if record locking rule is none or shared.

An overview of product features reveals a variety of capabilities. All of these products have security controls, but none have automatic deadlock recovery or transaction processing, which is extremely important for database integrity.

No specific capabilities are provided for transaction backout, logging, or rollback/rollforward for data recovery.

Security. Security in the LAN version is the same as in the single-user version. The user's name and password are required for login. Security levels are assigned by password to limit the user's access to the fields and the menus. A user file of names and passwords is encrypted for disk storage.

SORTING THROUGH FEATURES

Applying criteria is an essential part of understanding the multiuser capabilities of PC data managers. Evaluations of dBASE III PLUS, Paradox, and DataEase LAN, using such criteria, reveal wide variations in their multiuser approaches and capabilities (see table 4).

dBASE III PLUS was the first (by about two years) to implement multiuser features. Its data-sharing abilities are limited and restrictive in both interactive and programming modes. Its gray hairs are showing in its limited locking capabilities.

Paradox boasts a friendly, easy-to-use, menu-driven interface. It provides informative, data-sharing features for interactive users. It includes a variety of locking functions for programmers and implements intelligent locking defaults. It is a modern interactive data manager for LAN environments.

DataEase LAN introduces the concept of locking strategies for data sharing. Its plethora of locking options provides a vast assortment of possible combinations for data sharing in both interactive and programmed mode. Effective database administration, however, requires a comprehensive knowledge of DataEase processing methodology and a thorough understanding of data-sharing concepts.

All three data managers provide data security via passwords and access levels. None detects nor recovers from deadlock. Paradox's ability to lock sets of resources and DataEase LAN's variety of nonprocedural locking options can be helpful in avoiding deadlock. In dBASE III PLUS, deadlock avoidance necessitates the use of detailed programming constructs.

None of the products has full transaction-processing capabilities such as rollback, rollforward, and audit logging that are necessary for rapid recovery from the equipment and the program errors that frequently corrupt databases. Frequent backups and careful tracking of changes are required to ensure the continued integrity of the production databases. Although these products are useful in developing many important shared database applications, lack of formal transaction-processing capabilities is limiting. Automatic lock management might by itself seem easy and even foolproof; however, without automatic transaction processing, the burden of intelligent lock management still continues to be very much the responsibility of the user.

None of these data managers can, as yet, provide perfect multiuser capabilities. But data manager vendors are increasingly aware of the growing demand as well as the competition. They can be expected to continue making improvements in the multiuser features of their products.

Ashton-Tate
20101 Hamilton Avenue
Torrance, CA 90502-1319
213/329-8000
dBASE III PLUS 1.1
CIRCLE 336 ON READER SERVICE CARD

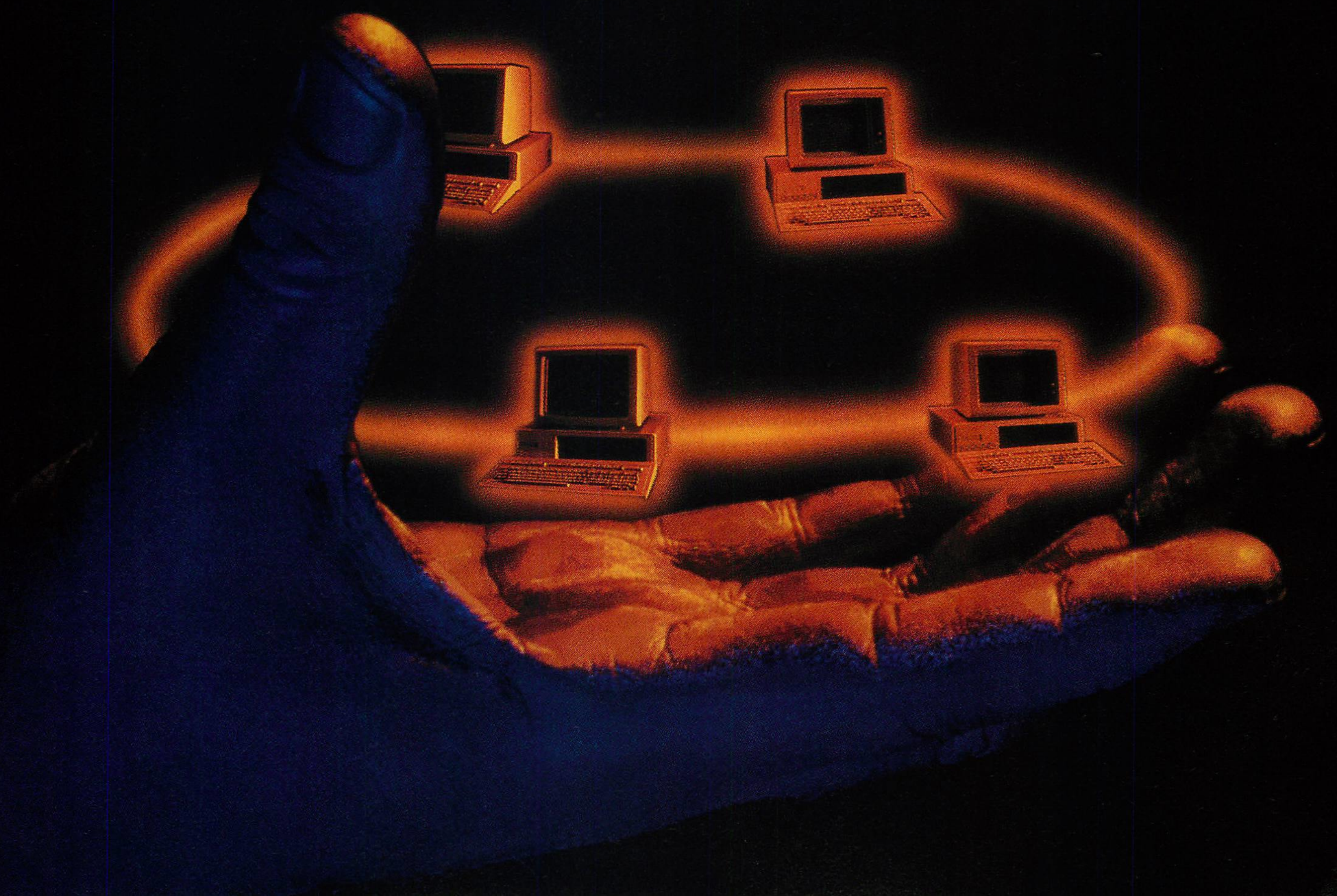
Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066-9987
408/438-8400
Paradox 2.0
CIRCLE 337 ON READER SERVICE CARD

DataEase International
7 Cambridge Drive
Trumbull, CT 06611
800/243-5123; 203/374-8000
DataEase LAN 1.02
CIRCLE 338 ON READER SERVICE CARD

Dave Browning is vice president and co-owner of WBS and Associates, Inc., a microcomputer and custom database consulting firm. He is also director of industry relations and chairman of the database special interest group for the Capital PC User Group.

The LAN Audit Trail

TOOLS FOR THE LAN ADMINISTRATOR



LAN auditors may not be foolproof security watchdogs, but they can provide valuable records of LAN operations for a variety of purposes.

ED SAWICKI

Novell Inc.'s network operating system, NetWare, is highly regarded for the reliability of its centralized file-server security. NetWare provides a diversity of file and directory access rights, making possible a spectrum of security configurations. An important security function not implemented to date by Novell is LAN access auditing—the facility to generate periodic reports that detail which file operations network users have performed on a file server.

Fortunately for LAN administrators, network audit utilities are being developed by third parties to enhance NetWare's security. These utilities record logins, logouts, and file operations of LAN workstations and produce reports on major categories of network activity.

Two such LAN audit utilities are LT Auditor from Blue Lance, and LANtrail from LAN Services, Inc. (LT Auditor was formerly known as LANTight; LANTight is now the name for Blue Lance's family of network administration products.) Both LT Auditor and LANtrail report on file operations performed by workstations accessing Novell file servers. LANtrail also reports on such network activity as changes to system rights, system errors, and disk use.

Although they are promoted by their developers as security software, LAN audit utilities also help in the development, integration, and diagnosis of LAN applications. The sophisticated user can disable or circumvent these audit software products, however, so LAN audit utilities may not be adequate for systems with mission-critical security needs. Even so, the assistance rendered for software debugging and LAN diagnostics may still justify the use of LAN audit products.

LAN audit utilities can improve security by monitoring which files specific users access. The audit software creates a log file on the server, which stores the audit information. The system administrator can use this file to identify the source of data corruption and illegitimate file access.

Without audit software, LAN administrators have three general types of security controls: physical access controls, system access controls, and application software access controls. (See "LAN Security," Art Krumrey, January 1988, p. 96.) These controls limit access by individuals who have no legitimate rights to the system or application. Unfortunately, they do not eliminate one of the greatest threats to the LAN's data. In many cases the people most likely to violate a system are disgruntled, dishonest, or inept employees.

LAN audit software addresses security issues for users with valid rights to the system. One of the drawbacks of audit security, however, is that the file I/O reporting is after the fact. The deterrence in this case lies in accountability. A classic form of access control, accountability relies on the threat of job loss or criminal prosecution for people who are entrusted with valuable data and allow theft or compromise.

Along with corrupting and deleting data, LAN sites with sensitive information should be concerned with data copied to local disk drives and taken off site. LAN audit utilities can detect copies from the network to a local disk drive and help control unwanted data transfers off the network.

AUDIT STRATEGIES

Regardless of which audit program is implemented, the LAN administrator should formulate a strategy for defining which specific file operations are to be audited and reported. LT Auditor can control the audit during the audit process, and LANtrail can target specific types of file access during the report-generation process. An audit strategy can be categorized according to a LAN's reporting needs, as follows:

High security. The high-security category requires a worst-case approach that assumes any file operation may be relevant to security reporting. All types of file operations should be reported, but it can be limited to directory areas that contain sensitive data.

Network-usage reporting. Audit reporting can assist in modeling network usage patterns. Usage auditing is restricted to monitoring file operations on major application program files. Reports generated from this type of audit can disclose which applications are used by specific users, when they are used, and the duration of use. This information helps LAN administrative activities such as capacity planning, cost justification, and software evaluation.

Software development and integration. Most LAN development and integration efforts can benefit greatly from audit software. In some cases it may be desirable to capture all file I/O to create a record of how test software accesses program and data files on the network. If debugging requires more specific information, audit reports can be limited to only certain types of file operations on a certain file. An example would be to report on file operations

that take place only in the application test directory, and only for files related to the application being tested.

LAN diagnostics. Systems management involves regular diagnoses of minor but potentially incapacitating faults. On even the best-designed LAN, a workstation can hang, resulting in possible loss of data and the need to reboot. Isolating faults on LANs is often difficult because of multiple factors, such as cable systems, distributed hardware, and complex software environments. If a LAN audit utility is in place when a fault occurs—and the hard disk is still intact—it is good trouble-shooting to analyze the audit report leading up to the failure. At minimum, the audit report should indicate which application the user was running at the time of the fault; ideally, audit software could provide an instant replay of the events that lead up to a network failure.

General-purpose auditing. This type of auditing records file operations for a number of users, across a wide range of applications. Because a general-purpose audit can record an enormous quantity of file operations, the audit should include a limited set of operations, directory areas, and file types. It is probably not necessary to record access of read-only public areas that contain DOS or Novell files. Also, it can be assumed that a file has been opened any time it is accessed for a read or write. Using this logic, all file opens can be excluded from reports. Certain applications repeatedly access their overlay files for common operations; these accesses need not be included in the audit.

If often-used batch files are included in audit profiles, the audit reports list one or more file operations for every line read as a batch file executes, making reports excessively long. For many applications, it is not desirable to audit for batch-file access.

EVADING AUDIT

A major vulnerability of LAN audit software is that it is possible for users to evade the Novell login procedure and thus escape auditing. Both LT Auditor and LANtrail load during the login process. If BREAK is ON, and Ctrl-Break is entered immediately after the login command is given, the user is logged in with normal rights, but with no path settings. Users sophisticated enough to know how to abort the login procedure are likely to know how to map drives. Once drives are mapped, the user can access files on the file server drives without being audited.

The LANtrail manual indicates that the user cannot log in to the network without the audit program being loaded. LANCHECK, the program that checks for initialization of the audit software, is loaded during the Novell login process and can be circumvented. A user also can disable the loading of the LT Auditor audit program.

Current audit software products conduct the audit process at the workstation; processes running on PCs are difficult to safeguard. If an audit utility running on the server could access audit information, however, this would not be a problem.

Audit reports can provide a fascinating look at the interface between application software and the network operating system.

Version 2.1 of SFT NetWare supports value-added processes (VAPs). A VAP is a network utility or application that resides in memory on a Novell file server and services requests from applications running on network workstations. An audit facility run as a VAP could reduce the possibility of users defeating the audit process.

A COMPLETE RECORD

Whether used for security purposes or analysis of network file operations, the end product of the LAN audit process is a series of reports detailing file accesses and security conditions. For seasoned network administrators and integrators, audit reports can provide a fascinating look at the interface between application software and the network operating system.

When using Novell network software, the usual method of observing network file operations is on the file-server console. The console monitor displays file requests in realtime. During periods of sustained file I/O, file request messages flicker by too fast to be fully comprehensible. The audit reports, however, display the network file operations in a lengthy chronicle that encourages careful inspection.

A typical report indicates when application programs are executed, configuration files are read, temporary files created, and overlays are opened and closed. See figure 1 for an access report of file activity monitored during user login, a WordPerfect session, and the LT Auditor supervisor utility. File operations are identified by the use of a single character: open a file (O), close a file (C), write to a file (W), erase a file (E), rename (R), create a file (T), make a directory (M), and delete a directory (D). Other reports generated by audit software summarize login activity, the history of security changes, and usage statistics.

Filtering the Audit. An important aspect of the network audit process is *filtering* (or narrowing the audit). Filtering is used when the LAN administrator has an interest in specific file activity or decides that a report of all file-access activity on the network is not needed in an audit report.

LANtrail and LT Auditor are at opposite ends of the spectrum in their philosophies of filtering, and each approach has significant tradeoffs. LT Auditor filters audit information at each PC workstation, and it can restrict the amount of data the user can collect and send to the audit log file. If filtering takes place at the workstation while the application is accessing files through DOS functions, the amount of network traffic generated by the audit shell can be reduced. (See the accompanying sidebar for a discussion of the audit shell mechanics.) The audit log file on the server grows less quickly with this method. LT Auditor lets the user specify a limit to the size of audit files and stops collecting audit data when available disk space falls below 1MB.

LANtrail, on the other hand, does its filtering when reports are printed. In this case, every file operation is trapped on the workstation and an unabridged record of all file activity is sent to the audit log file. Because all file access activity is collected, the log file is larger than if the filtering took place at the workstation. The LAN administrator can choose which file operations are to be listed in each report. For some applications, however, this type of filtering can produce a substantial amount of network traffic, and the audit log file on the server can grow very large. The benefit of LANtrail's method is that the audit data for all file operations are stored on the file server if needed for reporting in the future.

LANtrail has no protection against running out of disk space on the server. LAN Services recommends allocating a few megabytes of disk space to the audit function. It is the network administrator's responsibility to ensure

FIGURE 1: *LT Auditor File Access Report*

```
USER1    STN: 10005A00BBA9

01:44    [O        ] [ ]   net$log.dat    SYS:PUBLIC\
01:44    [ C       ] [ ]   net$log.dat    SYS:PUBLIC\
01:44    [O        ] [ ]   command.com    C:\
01:44    [ C       ] [ ]   command.com    C:\
01:44    [ C       ] [ ]   exit.bat       SYS:MENU\
01:44    [O        ] [ ]   exit.bat       SYS:MENU\
01:44    [O        ] [ ]   wp.exe         SYS:APP\WP\
01:44    [O        ] [ ]   {wp}sys.fil    SYS:APP\WP\
01:44    [ C       ] [ ]   {wp}sys.fil    SYS:APP\WP\
01:44    [O        ] [ ]   usr}sys.fil    SYS:DATA\USER\WP\
01:44    [ C       ] [ ]   usr}sys.fil    SYS:DATA\USER\WP\
01:44    [O        ] [ ]   usr}.chk       SYS:DATA\USER\WP\
01:44    [ C       ] [ ]   usr}.chk       SYS:DATA\USER\WP\
01:44    [   T     ] [ ]   usr}.tv1       SYS:DATA\USER\WP\
01:44    [     W  1] [ ]   usr}.tv1       SYS:DATA\USER\WP\
01:44    [O        ] [ ]   start.mac      SYS:DATA/USER/WP\
01:44    [ C       ] [ ]   start.mac      SYS:DATA/USER/WP\
01:44    [O        ] [ ]   file.005       SYS:DATA\USER\WP\
01:44    [O        ] [ ]   lex.wp         SYS:APP\WP\
01:44    [O        ] [ ]   usr}lex.sup    SYS:DATA\SHR\SUP\
01:44    [ C       ] [ ]   usr}lex.sup    SYS:DATA\SHR\SUP\
01:44    [ C       ] [ ]   lex.wp         SYS:APP\WP\
01:45    [O        ] [ ]   th.wp          SYS:APP\WP\
01:45    [ C       ] [ ]   th.wp          SYS:APP\WP\
01:45    [ C       ] [ ]   files.005      SYS:DATA\USER\WP\
01:45    [ C       ] [ ]   usr}.tv1       SYS:DATA\USER\WP\
01:45    [ C       ] [ ]   usr}.bv1       SYS:DATA\USER\WP\
01:45    [ C       ] [ ]   usr}.tv2       SYS:DATA\USER\WP\
01:45    [ C       ] [ ]   usr}.bv2       SYS:DATA\USER\WP\

01:45    [  E      ] [ ]   usr}?.???      SYS:DATA/USER/WP\
01:45    [O        ] [ ]   command.com    C:\
01:45    [ C       ] [ ]   command.com    C:\
01:46    [O        ] [ ]   lantight.exe   SYS:FILTER2\
01:46    [ C       ] [ ]   lantight.exe   SYS:FILTER2\
01:46    [O        ] [ ]   lan$msg.dat    SYS:FILTER2\
01:46    [O        ] [ ]   ibm$run.ovl    SYS:FILTER2\
01:46    [ C       ] [ ]   ibm$run.ovl    SYS:FILTER2\
01:46    [O        ] [ ]   net$log.msg    SYS:SYSTEM\
01:46    [   T     ] [ ]   temp4.dat      SYS:FILTER2/
01:46    [     W  1] [ ]   temp4.dat      SYS:FILTER2/
01:46    [ C       ] [ ]   net$log.msg    SYS:SYSTEM\
01:46    [     W 10] [ ]   temp4.dat      SYS:FILTER2/
01:46    [ C       ] [ ]   temp4.dat      SYS:FILTER2/
01:46    [  E      ] [ ]   temp4.dat      SYS:FILTER2/
01:46    [   T     ] [ ]   temp4.dat      SYS:FILTER2/
01:46    [   T     ] [ ]   sys$fil4.$$$   SYS:FILTER2\
01:46    [O        ] [ ]   sys$file.dat   SYS:FILTER2\
01:46    [     W  1] [ ]   sys$fil4.$$$   SYS:FILTER2\
01:47    [ C       ] [ ]   sys$file.dat   SYS:FILTER2\
01:47    [     W 14] [ ]   sys$fil4.$$$   SYS:FILTER2\
01:47    [ C       ] [ ]   sys$fil4.$$$   SYS:FILTER2\
01:47    [O        ] [ ]   sys$fil4.$$$   SYS:FILTER2\
01:47    [     W  1] [ ]   temp4.dat      SYS:FILTER2/
01:49    [   T     ] [ ]   lpt1           SYS:FILTER2\
01:49    [     W  1] [ ]   lpt1           SYS:FILTER2\
01:50    [     W 40] [ ]   lpt1           SYS:FILTER2\
01:50    [ C       ] [ ]   lpt1           SYS:FILTER2\
01:51    [ C       ] [ ]   sys$fil4.$$$   SYS:FILTER2\
01:51    [     W482] [ ]   temp4.dat      SYS:FILTER2/
```

The file access report details file operations on both network and local drives for one LAN user over a period of eight minutes. The report columns indicate (left to right) time of access, file operation, file name, and path to accessed file.

that the server does not run out of disk space. LANtrail audit software requires regular attention so that it does not adversely affect LAN operations. Despite log-file size, however, LANtrail offers greater flexibility on which reports are generated on user activity.

LT Auditor is suitable if it is known in advance what is to be audited, whereas LANtrail is effective if what is to be audited is not known in advance. With LT Auditor's approach, the audit log file contains a limited history of file access operations. LANtrail's log file, however, contains a complete historical record of file server access activity.

LT AUDITOR'S SMART SHELL

LT Auditor is highly versatile during the audit process because of its ability to control what it audits. It uses a shell to read a small filter module loaded at login. The module, which can be customized, regulates which file operations are recorded (see the accompanying sidebar). LT Auditor accomplishes this by performing the Novell WHOAMI function when it first loads. After the shell determines which user is logged in, it loads the appropriate filter module from the file server.

LT Auditor can audit the following operations: open file, close file, delete directory, erase file, make directory, rename file, create file, and write file. The audit shell filter can be configured to record all or any combination of these file operations. The current version of LT Auditor, 2.03, does not report file-read operations. Reads are inferred by an open-and-close operation without a write in between. For security and worst-case applications, it is sufficient to know that a file has been opened, but read reporting would be helpful for other types of auditing. As seen in figure 1, repeated file writes are "ganged" after the first occurrence.

The workstation shell itself consumes about 20KB of workstation memory. The filter module requires an additional 2KB or less of this memory.

In addition to the workstation shell, another program included with LT Auditor is the Supervisor Interface. This is a menu-driven utility that allows the administrator to customize the audit shells for each user. Menus for formatting and running reports are also accessed with the Supervisor Interface. The menu system that LT Auditor uses for the Supervisory Interface is built with the C-Worthy C language development system. The C-Worthy user interface, which works so well for the Novell utilities such as SYSCON and FILER, is somewhat confusing in the LT Auditor implementation.

The filter for each user's audit shell comprises one or more filter elements that describe specific files and file operations to be monitored. Each element requires the following parameters to be defined:

- **File Operation.** This parameter specifies which DOS file operations—such as opening or closing a file—will be audited for each filter element. File operations are identified in a filter element with a single character.
- **Path.** The path parameter can specify a specific directory to be audited and may also include a Novell volume name or drive identifier. LT Auditor is sensitive to current drive mappings and alerts the administrator if a path is not available when the filters are created. Fortunately, the filter still accepts the path with the assumption that the path will be available on the workstation when it runs. This filter option allows the administrator to target specific directory areas that may contain sensitive data.
- **File.** File names, or *filespecs*, with wild cards can be entered for the file parameter. This parameter instructs the shell to audit file operations for all files of a certain file name or extension. The file parameter works globally if the ALL command is specified, or for individual paths as deter-

mined by the path parameter. If the administrator is concerned only about access of certain files (for example, *.WK1), the file parameter will filter out all files except those with the given filespec. Each time a filter is modified, the workstation must be rebooted to reload the audit shell before the changes take effect.

- **Command.** The options for command are ALL or LOCAL. ALL works in conjunction with the path parameter and tells the shell to audit all subdirectories under the path specified. If no path is specified, all directories for all volumes are audited. If the LOCAL option is included in a filter element, all file operations on all local drives will be audited, regardless of paths.

The LT Auditor log file is indexed in a Btree format, which is updated as audit records are added to it. This significantly reduces the amount of time necessary to format a report; in fact, the LAN administrator can format and print a report in minutes. Large audit listings viewed with the screen report function can be scrolled rapidly from top to bottom. Blue Lance cites this speed of formatting and printing as one of the major benefits of LT Auditor.

In addition to reporting on file access activity, LT Auditor records each network login and logout and can generate an access report detailing user login activity and the time it occurred.

LT Auditor installation involves three phases. The first phase is automated and makes several subdirectories, copies files into them, changes file attributes, and makes additions to the trustee rights table for the group "Everyone." An install program that makes these changes on the file server can be unnerving to the LAN administrator, but LT Auditor does not have the option to make such changes manually.

AUDIT SHELL MECHANICS

Both LT Auditor and LANtrail conduct auditing of network file operations with a small TSR program or *audit shell.* This program loads into the workstation RAM, above the Novell NetWare shell in the memory map. When the program initially loads and executes, it redirects the PC's interrupt 21H pointer to itself. With the interrupts revectored, the audit shell can examine all DOS function calls and trap audit information.

Once examined, the DOS function calls are passed on to where the original interrupt 21H pointer indicated. If an intercepted function represents activity that must be audited, an audit record is then sent to a hidden audit log file on the server's hard disk. NetWare's workstation shell also uses this technique of intercepting DOS interrupt pointers. The NetWare shell receives the file requests after they are monitored by the audit shell and then forwards them to the file server for processing.

The process of an audit shell recording the opening of a file is demonstrated in the following steps.

1. The application performs a DOS INT 21H Open File Function. The audit shell traps the interrupt, detects the File Open function, and internally checks whether this function should be audited. If the call should be audited, the file operation data are stored temporarily.
2. The audit shell directs the interrupt to the NetWare shell, by pointing it to the address that was in the interrupt vector table prior to the audit shell being loaded. Throughout this procedure, the audit shell has made no modifications to the interrupt other than trapping it.
3. The NetWare shell now traps the interrupt and determines that it is a File Open function. NetWare then parses the filename string to isolate the drive letter and resolves whether the drive letter is for a network or a local drive. If the call was for a network drive, the shell directs the File Open function over the network to the NetWare operating system, which is managing file operations on the file server.
4. The file is now opened on the server. Confirmation of a successful open is sent back to the application over the same route that the function took on the way to the server. The audit shell has yet to send an audit record to the audit log file regarding the File Open function. The file I/O sequence up to this point is the same regardless of whether the audit shell is LT Auditor or LANtrail. From this point on, however, the procedure is very different for the two auditing products.

LT Auditor Shell. The LT Auditor shell is now ready to send a record to the audit log file on the server. The audit log file is a file named SYS$FILE.DAT and resides in the SYS:FILTER2\ directory. This file write is not buffered (as with LANtrail.) Therefore, there is contention for this file if two or more workstations try to write to it at the same time.

The audit shell manages contention by attempting to lock the SYS$FILE.DAT file. NetWare's semaphore locking is used. If the lock is successful, the audit shell opens the file, writes the audit record to the file, closes the file, and releases the lock. Each of these actions represents packet flow over the network, which impacts performance. If the audit shell is not successful at locking the SYS$FILE.DAT file, the shell must wait until it becomes unlocked. Then it will perform all of the file activity mentioned above.

LANtrail Audit Shell. In the case of the LANtrail shell, an audit record is built as an ASCII string. Included in the string is the pertinent audit data such as the function value for the open file request (3DH), the name of the file to be opened, time, date, station number, etc. This ASCII record is sent to the audit log file by calling NetWare's message-logging function.

The Novell function used by LANtrail is not much more complex than calling a DOS function: the AH register is loaded with E3H. The SI register is loaded with an address of a request buffer, and the DI register is loaded with an address of a reply buffer. The ASCII string is contained in request buffer. Now a software interrupt 21H is generated just as for a DOS function call, but E3H will vector to a Novell extended function call.

The Novell file server receives the audit record (ASCII string) and stores it in a temporary buffer. It will write the string to the NET$LOG.MSG file in the SYS:SYSTEM directory when directed to by the task manager. Because the network message-logging function call writes to a queued buffer, there is never contention for the audit log file. Furthermore, because the audit shells in all workstations only write to the NET$LOG.MSG file through the buffer and never read from it, there are none of the locking and synchronization problems normally associated with multiuser applications.

—*Ed Sawicki*

At one point in the automated portion of the installation, the administrator is asked to enter the default settings. The help-key message suggests reading the Configuration Options section of the manual for instructions. Unfortunately, this section is buried in an area of the manual that is not oriented toward installation and includes no introductory explanation of what the default settings do. Nomenclature for the default settings, such as "Auto display—Continuous diagnostics," can baffle the first-time user and do not match what is in the manual.

The configuration options for LT Auditor include:

- **File limit size.** Sets size in bytes to which the log file can grow before auditing is discontinued.
- **Display size/limit.** Displays the log-file sizes when the Supervisory Interface is invoked.
- **Continuous diagnostics.** Instructs the terminate-and-stay-resident (TSR) module to trap errors and post them to the NetWare message file that resides in the supervisor's SYS:SYSTEM directory. These error messages can be displayed automatically when the supervisor utility is run.
- **Environment diagnostics.** Tests the LT Auditor environment each time the Supervisory Interface invoked. This means LT Auditor will determine if its directories and file attributes are intact and if the user trustee rights to the LT Auditor program directories have been deleted or tampered with.
- **Auto archive.** Automatically archives the audit log file and the LAN-access log file when their size reaches the limit defined with the file limit size option. The default active file limit size is 5MB.

After the LT Auditor files are copied to the file server and the defaults set, the command to load the audit shell should be placed in the Novell login script. Several locations are suggested for inserting the audit shell command: the system login script, the private login script, and a batch file that is called by the login script with the EXIT command. Unfortunately, a sophisticated user can circumvent all of these methods and avoid loading the audit shell into the workstation.

When the command to load the audit shell has been inserted in the login sequence, a filter must be created by the supervisor utility for each user. Without this filter, file operations of the user workstation will not be recorded even if the audit shell is loaded. The supervisor utility creates the filter.

Basic installation is complete when a user's audit filter is created and the shell command is inserted in the login sequence. The audit process begins the next time a user logs in and the modified login script loads the audit shell into workstation memory.

The live audit log file that LT Auditor maintains cannot be exported directly. Thoughtfully included with LT Auditor is a utility for converting the audit log file into a flat ASCII format. The convert utility is not loaded automatically during the LT Auditor installation process; it is documented in a READ.ME file on the program diskettes.

The utility makes a complete pass of both the file-access and the LAN-access files and outputs new files with fixed-length formats. The file structure for a converted file-access audit file includes user name, station code, access date, access time, function call, file name, and path. The record length is 365 bytes. The file structure for a converted LAN-access audit file includes: user name, station code, login date, login time, login code, logout date, logout time, and logout code. Record length for LAN-access records are 97 bytes. The station code refers to the Novell physical address number.

LESS COMPLEX LANTRAIL

LANtrail version 2.0 provides an audit program with added modules for generating reports on a diverse range of network activities. LANtrail audits a slightly different set of file operations than does LT Audit. The following operations are monitored: open file, close file, check or modify date/time flags, write file, read file, and rename/create/delete files or directories.

LANtrail requires a svelte 5KB of memory, while LT Auditor takes up more than 20KB of the PC's conventional memory. The disparity in memory requirments is to be expected considering the LANtrail audit shell does not filter at the workstation and is a less complex piece of code.

LANtrail generates eight reports, detailing file, directory, and volume access as well as the activities of users, groups, and servers. Reports can be sent only to a local or network printer or file; unlike LT Auditor, they cannot be previewed on the screen.

Only users with supervisor equivalence can run reports. The following printers are supported: Hewlett-Packard LaserJet +, LaserJet + with landscape cartridge, Epson FX 100, Epson FX 80, HP LaserJet + 8, and HP LaserJet + 81. Other printers can be defined but the

administrator must know the exact printer codes. No control is available for the banner pages between reports.

LANtrail reports are designed to provide helpful information in a format appropriate for many LAN administration activities. The main product of the LANtrail audit process is the File-Access Report. It lists file operations for network and local drives as well as path, user name, station number, and time of access. The operations listed in this report are open, read, write, close, and date/time change. Failed file access attempts also are included in this report.

The Programs Executed Report lists the applications that network users access and helps usage tracking. This report includes physical station number, user name, time of access, application executable, and directory path (see figure 2). It targets a specified date range and is sorted by user or time.

The LANtrail Access Report details user access to the network, including users logged in during specified date range, server(s) logged into, physical workstation number, login time, and elapsed login time.

The Security Report shows the security status for each of the file server's active objects. This report displays object information, such as user IDs, passwords, group rights, security equivalences, group members, mailbox IDs, and directory/subdirectory trustee rights. System and individual login scripts can be printed. One version of this report is a single-page summary of all the rights and privileges for a specific user that gives an excellent summary of a single user's effective rights.

LANtrail's Bindery Changes Report specifies changes to the Novell Bindery within a date range. In NetWare, objects are defined as entities (such as users, groups, and servers) and are the basic units in the Novell Bindery system. The Bindery is a small, special-purpose database that forms the foundation for Novell's centralized security system. It is stored in hidden files in the SYS:SYSTEM directory of every Novell file server and is updated whenever security objects are modified.

The main elements of the Bindery Changes Report are security objects and properties (attributes of objects that may include passwords, internet addresses, or mail directories). This report provides its information using terms such as log in an object, create an object, change an object's security, and add properties to an object. All changes made to network security can be listed chronologically or by user name.

FIGURE 2. *LANtrail Programs Executed Report*

Monday December 14, 1987 — Programs Executed Report — for Server : LAB

Report for 12/1/87 to 12/13/87

USER NAME	STATION NUMBER	DATE	TIME	PROGRAM
USER1	[2 - 10005A00BBC4]	12/13/87	17:32	?:LANTRAIL\PROGRAMS\LANCHECK.COM
USER1	[2 - 10005A00BBC4]		17:32	?:LANTRAIL\PROGRAMS\LANCHECK.EXE
USER1	[2 - 10005A00BBC4]		17:33	LAB/SYS:PUBLIC\RIGHTS.EXE
USER1	[2 - 10005A00BBC4]		17:33	LAB/SYS:PUBLIC\RIGHTS.EXE
USER1	[2 - 10005A00BBC4]		17:33	LAB/SYS:PUBLIC\RIGHTS.EXE
USER1	[2 - 10005A00BBC4]		17:33	LAB/SYS:WP\WP.EXE
USER1	[2 - 10005A00BBC4]		17:34	LAB/SYS:PUBLIC\LOGIN.COM
USER1	[2 - 10005A00BBC4]		17:35	LAB/SYS:PUBLIC\LOGIN.COM
USER2	[2 - 10005A00BBA9]	12/11/87	12:31	?:LANTRAIL\PROGRAMS\LANCHECK.COM
USER2	[2 - 10005A00BBA9]		12:31	LAB/SYS:LANTRAIL\PROGRAMS\LANCHECK.EXE
USER2	[2 - 10005A00BBA9]		12:31	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		12:31	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		13:38	LAB/SYS:LOGIN\LOGIN.COM
USER2	[2 - 10005A00BBA9]		13:38	?:LANTRAIL\PROGRAMS\LANCHECK.COM
USER2	[2 - 10005A00BBA9]		13:38	LAB/SYS:LANTRAIL\PROGRAMS\LANCHECK.EXE
USER2	[2 - 10005A00BBA9]		13:38	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		13:38	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		13:39	C:MENU\XMENU.COM
USER2	[2 - 10005A00BBA9]		13:39	C:UTIL\FM.COM
USER2	[2 - 10005A00BBA9]		13:40	LAB/SYS:LANTRAIL\PROGRAMS\LANTRAIL.EXE
USER2	[2 - 10005A00BBA9]		13:45	C:UTIL\FM.COM
USER2	[2 - 10005A00BBA9]		13:48	C:UTIL\PE.EXE
USER2	[2 - 10005A00BBA9]		17:10	C:MENU\XMENU.COM
USER2	[2 - 10005A00BBA9]		17:23	LAB/SYS:DATA\USER2\LS.EXE
USER2	[2 - 10005A00BBA9]		17:27	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		17:27	LAB/SYS:APP\WP\WP.EXE
USER2	[2 - 10005A00BBA9]		17:28	LAB/SYS:PUBLIC\SYSCON.EXE
USER2	[2 - 10005A00BBA9]		17:30	LAB/SYS:PUBLIC\LOGIN.COM
USER2	[2 - 10005A00BBA9]		17:35	?:LANTRAIL\PROGRAMS\LANCHECK.COM
USER2	[2 - 10005A00BBA9]		17:35	LAB/SYS:LANTRAIL\PROGRAMS\LANCHECK.EXE
USER2	[2 - 10005A00BBA9]		17:35	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		17:35	LAB/SYS:PUBLIC\MAP.COM
USER2	[2 - 10005A00BBA9]		17:35	C:MENU\XMENU.COM
USER2	[2 - 10005A00BBA9]		17:36	C:UTIL\CURSOR.COM
USER2	[2 - 10005A00BBA9]		17:36	C:WP\WP.EXE
USER2	[2 - 10005A00BBA9]		17:37	C:MENU\XMENU.COM
USER2	[2 - 10005A00BBA9]		17:38	LAB/SYS:DATA\USER2\XTALK\XTALK.EXE
USER3	[2 - 10005A00BBB2]	12/13/87	17:30	LAB/SYS:PUBLIC\SYSCON.EXE
USER3	[2 - 10005A00BBB2]		17:30	LAB/SYS:PUBLIC\MAP.COM
USER3	[2 - 10005A00BBB2]		17:30	LAB/SYS:PUBLIC\FILER.EXE
USER3	[2 - 10005A00BBB2]		17:32	LAB/SYS:PUBLIC\LOGIN.COM
USER3	[2 - 10005A00BBB2]		17:35	LAB/SYS:PUBLIC\SYSCON.EXE
USER3	[2 - 10005A00BBB2]		17:35	LAB/SYS:LOGIN\LOGIN.COM
USER3	[2 - 10005A00BBB2]		17:43	LAB/SYS:PUBLIC\CHKVOL.COM
USER3	[2 - 10005A00BBB2]		17:43	LAB/SYS:PUBLIC\VOLINFO.EXE
USER3	[2 - 10005A00BBB2]		17:43	LAB/SYS:PUBLIC\MAP.COM
USER3	[2 - 10005A00BBB2]		17:44	C:DOS\FIND.EXE
USER3	[2 - 10005A00BBB2]		17:44	C:DOS\FIND.EXE
USER3	[2 - 10005A00BBB2]		17:44	C:DOS\FIND.EXE
USER3	[2 - 10005A00BBB2]		17:44	LAB/SYS:PUBLIC\LOGIN.COM

This report lists one entry for each access to a .EXE or .COM extension file. It provides a much more specific view of network activity than does the file access report in figure 1, which has many entries for each program executed.

The System-Error Log lists all system error messages and summarizes errors. It is more informative than the contents of Novell's error log file in the SYS:SYSTEM directory; this log includes the date and time of error and the frequency of error.

The Directory-Usage Report is a standard disk-usage summary that gives the number of files and bytes contained in each directory of the network hard disk(s). This report also calculates the percentage of the total volume that each directory occupies and the percentage of total files on the volume that each directory contains. Because it gives the subtotals for directories off the root directory, it is possible to determine the total number of bytes in a user's home directory and all its subdirectories in a single glance. This aids in hard-disk capacity planning.

FIGURE 3. *LANtrail Rename, Create, Delete Report*

Monday December 14, 1987 — Rename, Create, Delete Report — for Server : LAB

Report for 12/ 1/87 to 12/13/87

USER NAME	STATION NUMBER	DATE	TIME	FILES/DIRS	UNIQUE FILE NAME
USER1	[2 - 10005A00BBA9]	12/13/87	17:33	[C]	LAB/SYS:DATA\TESTDOC
USER1	[2 - 10005A00BBA9]		17:33	[C]	LAB/SYS:WP\USR}.CHK
USER1	[2 - 10005A00BBA9]		17:33	[C]	LAB/SYS:WP\USR}.TV1
USER1	[2 - 10005A00BBA9]		17:33	[C]	LAB/SYS:WP\USR}.BV1
USER1	[2 - 10005A00BBA9]		17:33	[C]	LAB/SYS:WP\USR}.SPC
USER1	[2 - 10005A00BBA9]		17:34	[C]	LAB/SYS:WP\MEMO3
USER1	[2 - 10005A00BBA9]		17:34	[C]	LAB/SYS:WP\MEMO5
USER1	[2 - 10005A00BBA9]		17:34	[C]	LAB/SYS:WP\MASTER.PLN
USER1	[2 - 10005A00BBA9]		17:34	[D]	LAB/SYS:WP\MEMO3
USER1	[2 - 10005A00BBA9]		17:34	[D]	LAB/SYS:WP\MEMO5
USER1	[2 - 10005A00BBA9]		17:34	[D]	LAB/SYS:WP\MASTER.PLN
USER1	[2 - 10005A00BBA9]		17:34	[D]	LAB/SYS:WP\USR}?.???
USER2	[2 - 10005A00BBA9]	12/11/87	22:31	[D]	LAB/SYS:DATA\USER2\RUNIT.BAT
USER2	[2 - 10005A00BBA9]		22:31	[C]	LAB/SYS:DATA\USER2\RUNIT.BAT
USER2	[2 - 10005A00BBA9]		22:33	[C]	LAB/SYS:MENU\S.BAT
USER2	[2 - 10005A00BBA9]		22:33	[C]	C:{PE}.PEC
USER2	[2 - 10005A00BBA9]		22:33	[C]	C:{PE}1.PET
USER2	[2 - 10005A00BBA9]		22:33	[C]	C:{PE}1.PEB
USER2	[2 - 10005A00BBA9]		22:33	[C]	C:{PE}.PES
USER2	[2 - 10005A00BBA9]		22:33	[R]	C:MENU.BAT
USER2	[2 - 10005A00BBA9]		22:33	[D]	C:{PE}????.PE?
USER2	[2 - 10005A00BBA9]		22:34	[C]	C:{WP}.CHK
USER2	[2 - 10005A00BBA9]		22:34	[C]	C:{WP}.TV1
USER2	[2 - 10005A00BBA9]		22:34	[C]	C:{WP}.BV1
USER2	[2 - 10005A00BBA9]		22:34	[C]	C:{WP}.SPC
USER2	[2 - 10005A00BBA9]		22:36	[D]	C:{WP}?.???
USER2	[2 - 10005A00BBA9]		22:37	[C]	C:MSG00003
USER2	[2 - 10005A00BBA9]		22:37	[C]	C:MSG00004
USER2	[2 - 10005A00BBA9]		22:37	[C]	C:MSG00006
USER2	[2 - 10005A00BBA9]		22:37	[C]	C:MSG00007
USER2	[2 - 10005A00BBA9]		17:45	[C]	LAB/SYS:LANTRAIL\PROGRAMS\PRN
USER2	[2 - 10005A00BBA9]		17:49	[R]	LAB/SYS:SYSTEM\NET$LOG.MSG
USER2	[2 - 10005A00BBA9]		17:49	[C]	LAB/SYS:LANTRAIL\DATA\WHOSWHO.TMP
USER2	[2 - 10005A00BBA9]		17:49	[D]	?:WHOSWHO.BAK
USER2	[2 - 10005A00BBA9]		17:49	[R]	LAB/SYS:LANTRAIL\DATA\WHOSWHO.TMP
USER2	[2 - 10005A00BBA9]		17:51	[C]	LAB/SYS:LANTRAIL\DATA\FILEACC.TMP
USER2	[2 - 10005A00BBA9]		17:51	[D]	LAB/SYS:SYSTEM\NET$LOG.TMP
USER2	[2 - 10005A00BBA9]		17:51	[D]	LAB/SYS:LANTRAIL\DATA\WHOSWHO.BAK
USER2	[2 - 10005A00BBA9]		18:32	[R]	LAB/SYS:SYSTEM\NET$LOG.MSG
USER3	[2 - 10005A00BBA9]	12/13/87	17:31	[C]	LAB/SYS:PUBLIC\DRIVERS
USER3	[2 - 10005A00BBA9]		17:31	[C]	LAB/SYS:PUBLIC\LISTINGS
USER3	[2 - 10005A00BBA9]		17:31	[C]	LAB/SYS:PUBLIC\REPORTS
USER3	[2 - 10005A00BBA9]		17:31	[D]	LAB/SYS:PUBLIC\DRIVERS
USER3	[2 - 10005A00BBA9]		17:31	[D]	LAB/SYS:PUBLIC\LISTINGS
USER3	[2 - 10005A00BBA9]		17:31	[D]	LAB/SYS:PUBLIC\REPORTS
USER3	[2 - 10005A00BBA9]		17:32	[D]	LAB/SYS:SYSMAN\????????.???
USER3	[2 - 10005A00BBA9]		17:32	[D]	LAB/SYS:SYSMAN

Reports on file and directory modifications can be sorted chronologically in the order they occurred, or they can be grouped by user name as in this report. Abbreviations in the Files/Dirs column are: C (create), D (delete), and R (rename). This report lists operations for three users on local and network drives.

The Rename, Delete, and Create Report lists modifications to files and directories on given date or range of dates. It includes information on server name, user name, station number, time of modification, and file name and/or directory path. It can be sorted by time or user (see figure 3).

Before a report can be printed, the audit log file must go through a process known as compression, which greatly reduces file size. In one test, the compress procedure for an audit log file took about 1 hour 30 minutes. This audit file was produced by capturing all file operations for three network stations engaged in word processing and light database management for an eight-hour period. The compression procedure delays the report generation considerably; compression is not necessary for LT Auditor.

LANtrail can run the report as a batch job. LAN Services recommends running the reports on a fast PC or at night when time does not matter. Novell server printers 0-4 and one local printer (LPT1) are easily selected from the PRINTER SET option within the main LANtrail menu.

Installation of LANtrail is accomplished automatically by running the INSTALL program from the program diskette. A LANtrail directory is created with two subdirectories: PROGRAMS and DATA. The PROGRAMS subdirectory contains all of the program files; the DATA subdirectory contains all of the compressed report files.

LANTRAIL.COM, the TSR audit shell, is placed into the Login subdirectory. LANtrail must be called before the LOGIN command. If LANTRAIL.COM is not loaded, an error message will ask the user to reboot.

NET$LOG.MSG is the file placed in Novell's SYS:SYSTEM directory where all audit information is initially stored. The COMPRESS utility sorts the information into the appropriate report files in the \LANtrail\DATA directory.

The administrator is directed to map a drive to the LANtrail program subdirectory for convenience.

LANtrail can control the number of simultaneous logins by a user. The LANCHECK program, which is loaded during the login process, checks for concurrent logins under the same login name. The numeric argument for this program sets the number of sessions that any user can have at the same time. This program can be placed in the system login script if it is to apply to all users or in the private login script to limit an individual user.

When an illegal session is attempted, a message informs the user that all of the available sessions are already in use. At this point the workstation hangs and must be rebooted.

During the installation process, LANtrail adds the following lines to the beginning of the system login script:

Break off
DOS Break off
IF "LOGIN_NAME" < >"SUPERVISOR" THEN BEGIN
#SYS LANtrail \PROGRAMS\LANCHECK
END

LANtrail protects the audit log file from users by writing audit records in NET$LOG.MSG in the SYS:SYSTEM directory. The function call allows only writes to the file; these writes are appended to the end of the file.

The only way to read or erase the audit log file is to have rights in the SYSTEM directory. End users normally

do not have access rights to this directory, so they cannot overwrite the audit information. However, a user could conceivably create a program that writes to the NET$LOG.MSG file. This could destroy the log file, and LANtrail would then have nothing with which to produce reports.

PERFORMANCE DIFFERENCES

LT Auditor's workstation filtering has the advantage of reducing the number of audit records that traverse the network cable and are written to the audit log file on the server. LANtrail does not filter at the workstation so it potentially sends more data over the network.

Performance differences between the two products may not be noticed if the network is small or the application does not perform frequent file I/O. The server's ability to service file requests will be reduced by higher audit activity, although word processing or spreadsheet users should not be affected. Applications with more ambitious file I/O requirements, such as data managers, are likely to make the performance degradation more obvious.

Several tests were run to measure the rate of audit log-file growth and the impact of auditing on network application performance. The performance tests were implemented on 80286 and 80386 workstations configured for the IBM Token-Ring Network. The file server was a 16-MHz Compaq Deskpro 386 with an internal 130MB drive. Novell NetWare version 2.0A was run on the file server Deskpro, also configured for Token-Ring topology.

Benchmarking was conducted with Innovative Software's multiuser application suite, the SmartLAN Performance Test. This software simulated a moderate-to-heavy-demand application environment of word processing, spreadsheet, and database processing. A station performing 1,000 writes to a database file was timed. All stations under test ran the audit software. File server utilization during the test activities (as shown on the Novell status screen) ranged from 0 to 30 percent.

As the benchmarks in table 1 indicate, LT Auditor has a distinct advantage over LANtrail in performance and control of audit file size when filtering takes place at the workstation. Even if all file operations are monitored, the performance advantage stays with LT Auditor by a small margin.

Log-file figures in table 1 are for the number of bytes recorded in the log file by all stations during each test. The log file clearly demonstrates that

TABLE 1: ***Audit Software Performance Benchmarks***

	NO AUDIT	BLUE LANCE		LAN SERVICES
PRODUCT		**LT Auditor**		**LANtrail**
		Filtered	Not Filtered	
TIME				
1 Station	0:55	0:57	0:58	1:01
2 Stations	0:56	0:58	0:59	1:02
4 Stations	0:57	0:59	1:01	1:03
6 Stations	0:58	1:01	1:05	1:05
LOG-FILE SIZE				
1 Station	N/A	1.3	6.6	6.1
2 Stations	N/A	3.7	15.3	16.7
4 Stations	N/A	9.3	37.0	35.4
6 Stations	N/A	15.3	59.0	59.3

N/A = Not applicable
All times in minutes:seconds
Log-file size in kilobytes

The performance test measured elapsed times for 1,000 database writes while other workstations were running applications. The No Audit column, which lists test times for one to six network workstations with no audit software loaded, represents the control group. The LT Auditor tests were run with no filtering and almost complete filtering, allowing only writes and creates to be audited.

the audit process can generate an enormous amount of data. In the case of no filtering, six workstations running either LANtrail or LT Auditor added almost 60KB of data to the audit log file in less than 10 minutes. It should be noted that all of these stations were engaged in fairly heavy activity. LANs with limited disk storage resources may be overwhelmed by the data collected with an unfiltered audit.

UNRESOLVED VULNERABILITIES

Although LAN audit utilities such as LT Auditor and LANtrail may not be suitable for security reporting in very high-security environments, their use as debugging and diagnostic tools make them helpful additions to the LAN administrator's toolbox.

LAN audit software can enhance Novell NetWare's intrinsic security with useful reports on file access, without imposing significant levels of overhead for many applications. Before deciding on such auditing software, prospective users should run their performance-sensitive, I/O bound applications, both with and without the audit shell in place, to evaluate degradation. This is particularly important for older 8088 machines running at 4.7 MHz; these machines suffer a higher level of degradation as compared to 286 or 386 machines.

Perspectives gained from such quantitative evaluations should be supplemented with the subjective observations of end users who are accustomed to normal application response times. Because audit shells are active only during file operations, they should have no effect on CPU-intensive processing local to the workstation.

Under NetWare 2.0, there is no guarantee that an audit shell will be loaded into the network workstation. As with the advanced security available on large centralized systems, LAN auditing should be closely tied to the operating system. The drawbacks of LT Auditor and LANtrail are largely related to their independence from NetWare's processes on the file server. Until this vulnerability is resolved, audit software will be at best a partial safeguard against the compromise of sensitive data on LANs.

Blue Lance
P.O. Box 430546
Houston, TX 77243
713/680-1187
LT Auditor 2.03: $495 per server
CIRCLE 340 ON READER SERVICE CARD

LAN Services Inc.
19 Rector Street
15th Floor
New York, NY 10006
212/797-3800
LANtrail 2.0: $595 per server
CIRCLE 341 ON READER SERVICE CARD

Ed Sawicki is director of the Accelerated Learning Center in Lake Oswego, Oregon, and lectures throughout North America and Europe on PCs and local area networks.

How to Buy the Right Monitor.

Let's cut through the hype and misinformation about large-screen monitors for desktop publishing. We've put together a set of tough questions to ask your dealer. Here are a few of them:

How fast can it update the screen?

Moving text or graphics around on a large-screen monitor means updating one million or more pixels. That can take forever on some monitors. Insist on a high-speed, built-in graphics coprocessor, like the Viking 1 controller board. Speed equals productivity!

What's it compatible with?

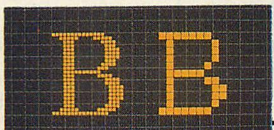
Protect your investment. Make sure your monitor can work on a variety of computers and software. The Viking 1 works with the IBM PC, PS/2, Mac SE, and Macintosh II. And with Aldus PageMaker, Ventura Publisher, Windows 2.0, WordPerfect, Lotus 1-2-3 and most word processing and CAD/CAM programs. Now *that's* compatibility!

How big is the screen?

A 15-inch or 17-inch monitor shows you only part of a page. You'll quickly get frustrated scrolling and zooming. A 19-inch monitor is required for real page layout, especially for facing pages. We even offer a 24-inch monitor! The Viking 2400—for the ultimate in page layout productivity.

How WYSIWYG is it?

Eliminate guesswork. Your screen fonts should exactly match your printer's fonts. Bitstream® Fontware™ creates matching screen and printer fonts for use with Ventura Publisher and Aldus PageMaker. Moniterm is the only monitor company that bundles Fontware with every monitor. Free.

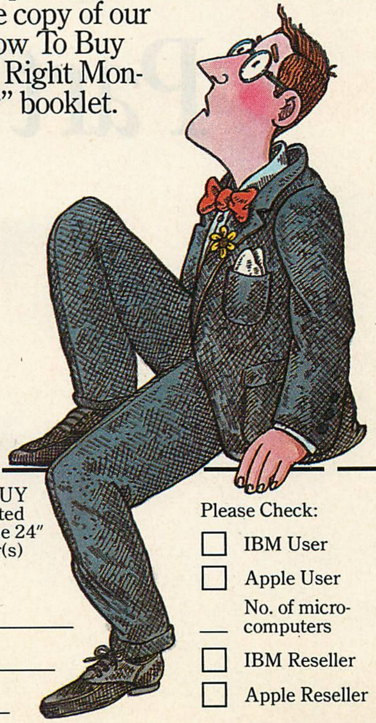


BITSTREAM® FONTWARE™ STANDARD FONT

What are the ergonomics?

Avoid screens that can cause headaches and eyestrain. Check for fuzzy fonts and screen reflections. (We offer optional non-glare filters.) Check for flicker (caused by "interlacing" and refresh rate lower than 63 Hz). Moniterm offers high refresh, super-crisp non-glare monitors. That's comfort!

To get the complete story—including simple monitor tests you can perform—write or call for your free copy of our "How To Buy the Right Monitor" booklet.



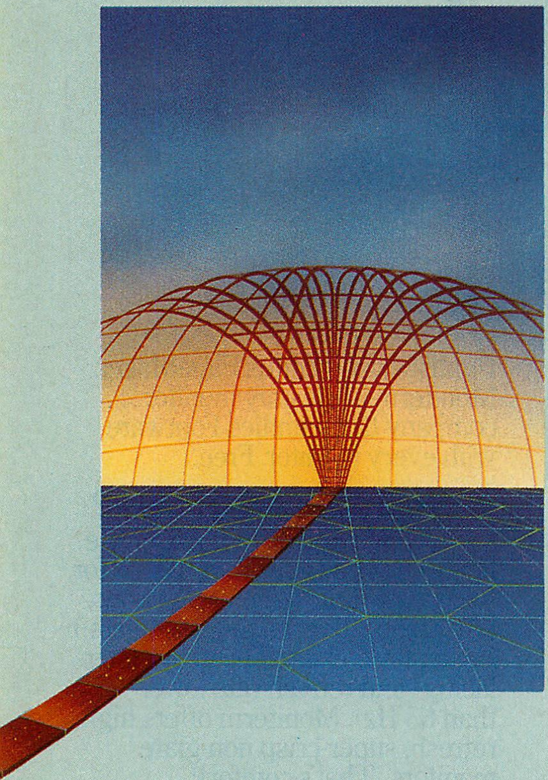
MONITERM

5740 GREEN CIRCLE DRIVE
MINNETONKA, MN 55343
(612) 935-4151
FAX # (612) 933-5701
TELEX #753626

Yes! Send me a *free* copy of HOW TO BUY THE RIGHT MONITOR. I am interested in ☐ monochrome 19″ ☐ monochrome 24″ ☐ color 19″ ☐ gray scale 19″ monitor(s) for desktop publishing!

Name ______
Title ______
Company ______
Address ______
City, State, Zip ______
Telephone ______

Please Check:
☐ IBM User
☐ Apple User
___ No. of microcomputers
☐ IBM Reseller
☐ Apple Reseller



Designing Drivers for OS/2

Part 2

ILLUSTRATION • ROBERT PASTERNAK

Device drivers for OS/2 must be able to fulfill certain special conditions that stem from the very nature of this new operating system: they must be able to function in both real and protected modes, allow multitasking (which DOS does not), and must be written down to the hardware level. In these areas, the applications developer's task can be simplified by Device Helper (DevHelp) services.

Part 1 of this article ("Designing Drivers for OS/2," David A. Schmitt, December 1987, p. 164) described the architecture of OS/2 device drivers, compared it with that of DOS drivers, and introduced the DevHelp services. This second installment examines the DevHelp services in greater detail, indicating how they simplify the development of I/O drivers in the complex OS/2 environment. The design and implementation of an actual character device driver are then described.

The DevHelps are a set of specialized system functions designed for use by device drivers only. Like application processes, device drivers need access to system-level services for controlling multitasking, memory allocation, and interprocess communications. However, drivers are not stand-alone processes, but execute as part of another process. Except for a subset of functions they can call during initialization, they do not have access to the wealth of operating system services through the normal applications program interface (API) (see "The Flexible Interface," David A. Schmitt, November 1987, p. 110). Instead, the DevHelp services provide duplicates of those API functions that are useful to drivers, plus some additional ones that are driver-specific.

Like the normal API, the DevHelp services are a collection of functions reached via far calls. But the DevHelp functions are numbered, not named,

Applications developers can turn to OS/2's Device Helper services for guidance through the new, complex environment in which they must design drivers.

DAVID A. SCHMITT

and the calling protocol, unlike that for the normal API, is designed for assembly language, not for high-level languages. It is, in fact, reminiscent of the DOS API protocol, in that all of the functions are reached through one address with the function number in a register. The driver gets the address of the DevHelp entry point from the request packet passed by the OS/2 kernel during initialization. The protocol for calling DevHelps is given below.

- Load the DevHelp function code into the DL register.
- Load the other arguments into registers, as appropriate for the function that is being called.
- Execute an indirect far call to the DevHelp entry address that is saved during initialization.
- On return, if the carry flag (CF) is set, then an error has occurred, and AX contains an error code.
- If CF is clear, then the operation was successful, and the registers contain any information that is returned by DevHelp services.
- All registers not used for return information are preserved.
- Most DevHelp functions preserve the state of the interrupt enable flag, except for those which may yield control of the CPU to another thread. They return with interrupts enabled.

HELP IN MANY FORMS

The DevHelp services are listed by functional groups in table 1 (for a tabulation by function code, see part 1, p. 180). The OS/2 *Technical Reference* provides complete programming information for using these in drivers.

Memory management services. Proper management of driver memory access is the most difficult aspect of OS/2 driver design and can be the source of many obscure bugs. This is because drivers must be bimodal—that is, able to operate in either real or protected mode. Furthermore, drivers must be prepared to begin an I/O operation in one mode and complete it in another, because a mode switch can occur between the time the strategy routine initiates the operation and the interrupt routine completes it. Whereas many of the DevHelp functions are useful but optional, the ones that manage memory access are mandatory because they manipulate the protected-mode segment descriptor tables at the heart of OS/2 (for details see the sidebar, "How Protected Mode Protects," Ted Mirecki, November 1987, p. 80).

Regardless of mode, the driver can always access its own code and data segments because the kernel sets up the CS and DS registers properly. However, pointers saved in memory will become incorrect when the mode changes. Also, any pointers into the caller's address space need to be converted into a form suitable for the current mode, and the segments to which they refer need to be mapped into the driver's address space.

Because of this bimodality requirement, drivers must save pointers in a representation known as a *physical address*. This is simply a 32-bit value (actually, a 24-bit value in a double word) identifying an absolute location in the physical memory space, which is 16MB on the 80286. For actually addressing memory, a physical address must be converted into a *virtual address* that is appropriate for the current mode: a segment: offset pair in real mode and a selector: offset pair in protected mode.

For several memory areas that are referenced very frequently by the driver, OS/2 provides *bimodal addresses* that can be used in either mode without conversion. Such addresses are provided for the BIOS data area at absolute location 400H, for the driver request packet passed to the strategy routine, and for the global system information area (described under the GetDOSVar function below). A bimodal address is created by setting up a descriptor table in such a way that the segment and selector values are identical. For example, the descriptor entry defining the memory for the BIOS data area is placed at offset 40H of the appropriate descriptor table, so that the value 40H can be used as a segment address in real mode and as a selector value in protected mode.

But the driver receives most addresses, most notably pointers to data buffers in the caller's address space, in virtual format that is valid only in whatever mode is in effect when the strategy routine is called. The driver uses the DevHelp function VirtToPhys to convert a virtual address into its physical equivalent, which can be saved across a mode switch. Before calling this function, the driver must lock the segment containing the virtual address if it is not already locked. This prevents the segment from being swapped or relocated in physical memory.

PhysToVirt performs the complementary operation of converting a physical address to the appropriate virtual form. Besides a double-word physical address, parameters to this function are the length of the memory block beginning at this address, and a switch that determines whether the virtual address is returned in DS:SI or ES:DI. This function performs various tasks, depending on the location of the physical address being converted and on the current addressing mode.

The simplest case is when PhysToVirt is called from real mode with a physical address below 1MB. It then returns a virtual address in segment: offset form, with the zero flag (ZF) clear to indicate that the addressing mode did not change.

Somewhat more complex but still fairly straightforward is the case when PhysToVirt is called from protected mode. It maps the requested memory block into the driver's protected address space by inserting into the appropriate memory mapping table a descriptor defining the physical origin and length of the block. The function returns with an address in selector: offset format in the requested register pair and with ZF clear to indicate that there was no mode change.

The most complex situation occurs when PhysToVirt is called from real mode with an address above 1MB; then its action depends on the hardware configuration of the system. If the system has an 80386 processor or an 80286 that supports the fast mode-switch feature (such as the IBM PS/2 Models 50 and 60), it is switched into protected mode, and the specified physical block is mapped into the driver's protected address space as described above. The virtual address is returned in selector:offset format in either DS:SI or ES:DI, as specified. The other data segment register is automatically converted if it currently refers to the driver's data segment; otherwise, it is unchanged. CS is also converted to the proper form. ZF is set to indicate that a mode change occurred.

Upon return, all previously stored virtual addresses except bimodal ones become invalid, and the driver must regenerate them by converting their physical forms before use. Therefore, any pointers not saved as physical addresses become unusable until the switch back to real mode.

If the system does not support fast mode switching (for example, the IBM AT), OS/2 makes use of the undocumented 80286 instruction LOADALL that

TABLE 1: ***Device Helper Services***

NAME	FUNCTION
MEMORY-ALLOCATION SERVICES	
AllocPhys	Allocate physical memory segment
FreePhys	Free physical memory segment
Lock	Lock memory segment
Unlock	Unlock memory segment
PhysToVirt	Convert physical to virtual address
PhysToUVirt	Convert physical to user virtual address
UnPhysToVirt	Release virtual address use
VirtToPhys	Convert virtual address to physical
VerifyAccess	Verify memory access is legal
REQUEST QUEUE SERVICES	
AllocReqPacket	Allocate memory for request packet
FreeReqPacket	Free allocated request packet
PushReqPacket	Add request packet to tail of queue
PullReqPacket	Get request packet from head of queue
PullParticular	Get a specific request packet from queue
SortReqPacket	Insert packet into queue in sorted order
PROCESS MANAGEMENT SERVICES	
Block	Suspend driver execution
Run	Awaken driver task
Yield	Relinquish CPU to another thread
TCYield	Relinquish CPU to time-critical thread
SendEvent	Notify kernel of special keystrokes
DevDone	Indicate I/O is complete
SEMAPHORE SERVICES	
SemHandle	Create/release a system semaphore
SemRequest	Test and set a semaphore
SemClear	Clear a semaphore
CHARACTER QUEUE SERVICES	
QueueInit	Initialize queue structure
QueueWrite	Insert character at tail of queue
QueueRead	Read character from head of queue
QueueFlush	Remove all characters from queue
INTERRUPT SERVICES	
SetIRQ	Hook hardware interrupt vector
UnSetIRQ	Unhook hardware interrupt vector
SetROMVector	Hook software interrupt vector
EOI	Issue end-of-interrupt
ROMCritSect	Enter/leave noninterruptible code
TIMER SERVICES	
SchedClockAddr	Get address of system timer routine
SetTimer	Register timer-interval handler
ResetTimer	Remove timer-interval handler
TickCount	Set timer interval
MONITOR SERVICES	
MonCreate	Create empty monitor chain
Register	Add a monitor to the chain
DeRegister	Remove monitor from the chain
MonWrite	Pass data to the monitor chain
MonFlush	Flush data from monitor chain
MISCELLANEOUS SYSTEM SERVICES	
GetDOSVar	Get system information
GetLIDEntry	Get ABIOS logical ID
FreeLIDEntry	Release ABIOS logical ID
ABIOSCall	Call ABIOS procedure
ABIOSCommonEntry	Call ABIOS common entry point

Some DevHelp services, like memory management, are indispensable to driver operation; others, like queue management, are merely conveniences.

allows access above the 1MB boundary in real mode. This is a diagnostic aid used by Intel during manufacture to test the chip's ability to address its entire address space. This technique puts the chip into an unstable state and works only as long as certain rules are obeyed. The contents of the segment registers must not be changed in any way, such as by saving and restoring them. Because interrupt handlers save and restore registers, PhysToVirt disables interrupts when it switches the CPU into this state, and the driver must not enable them until it has finished using the virtual address.

To optimize performance when converting several physical addresses, the address that is most likely to cause a switch into protected mode is converted first (that is, the one above the 1MB boundary). Also, to ensure correct operation of the driver on both fast-switch and slow-switch machines, the following rules are observed after PhysToVirt is called:

- Do not make other DevHelp calls, apart from a second PhysToVirt call.
- Do not attempt to PUSH/POP or save/restore a segment register that contains a virtual address.
- Do not modify a virtual address or save it for later use.

Observing these rules, the driver can handle only two virtual addresses at a time, one in DS:SI and one in ES:DI, which rarely is a serious restriction in practice.

Properly converted addresses remain valid until the driver returns to its caller or relinquishes control to the kernel via functions such as Block, Yield, TCYield, SemRequest, MonWrite, or EOI. Before calling any of these DevHelp services, the driver must release the memory management resources used for these converted addresses by calling UnPhysToVirt. This function need be called only once, regardless of how many physical addresses were converted. If called from protected mode, it removes from the memory mapping tables any descriptors created by one or more previous PhysToVirt calls. If any of these previous calls caused a mode switch, UnPhysToVirt restores the previous addressing mode and returns with ZF set. This alerts the driver to recalculate stored virtual addresses before reuse.

PhysToUVirt converts a physical address below the 1MB boundary into a suitable virtual form. It is more efficient than PhysToVirt and is normally used to access fixed locations in low memory or in the ROM BIOS area. When PhysToUVirt is called with DH set to a value of 2, it releases any resources allocated as a result of previously converting the specified address. UnPhysToVirt need not be called.

Memory-access control involves several other important functions besides address conversion. VerifyAccess checks if the user process is allowed to access the specified memory block. This function must be used to verify addresses passed by the user in request packets other than the normal read and write operations. The kernel does the verification for read/write requests, and it also locks the read/write area into physical memory. If it is not verified that the user process can legitimately access a segment, the driver could mutilate memory owned by some other process. If this function determines that the calling process does not have access to the memory block, the process is terminated. This verification is performed in protected mode only; access is always allowed in real mode.

Because the lower megabyte of physical memory is a fairly precious resource, the driver should normally ask for high memory.

After calling VerifyAccess to ensure that you are allowed to access the segment, you should then call Lock. Locking a segment guarantees that it will remain in the same physical memory area until unlocked. It is important that the driver not relinquish the CPU between the VerifyAccess and the Lock, because the access privilege may change if the driver takes a break. Once the segment is locked, VirtToPhys can be called to obtain its physical address. When access to the segment is no longer needed, Unlock is called.

One of the Lock parameters specifies whether the lock is short-term or long-term. Short-term locks are more efficient and should be used if the segment is to be unlocked within a few seconds. When a long-term lock is requested, OS/2 may move the segment to a different place in memory so that it will not get in the way of other segments that can be swapped.

AllocPhys provides the means by which the driver can acquire additional data space either above or below the 1MB boundary. This function returns a physical address that must be converted via PhysToVirt before it can be used. The space allocated should not be locked or unlocked, because it is fixed and will remain allocated to the driver until FreePhys is called. Because the lower megabyte of physical memory is a fairly precious resource, the driver should normally ask for high memory. If that request fails, then it should try allocating in the lower area.

Any space obtained via AllocPhys remains assigned to the driver until it is released via FreePhys. If you forget to release such areas when you receive a deinstall request, the memory will be lost to the system.

Request queue services. All driver operations begin with a request packet received from the OS/2 kernel as a result of a request from a user process. A request packet is simply a variable-length data structure that begins with a 13-byte header, as shown in figure 1. At entry to the strategy routine, ES:BX contains the far address of the request packet constructed by the kernel. This address is bimodal.

Drivers often place request packets on various queues as the I/O operation progresses, and DevHelp includes several functions to simplify request queue management. Use of these services is optional; a driver may perform its own queue management or implement a different data structure for storing I/O requests.

A standard request queue begins with a double-word pointer in the driver's data area that holds the far address of the first request packet in the queue. Within each enqueued packet, the device queue linkage field points to the next packet in the queue. The head pointer is zero when the queue is empty, and the last packet in the queue has zero in its linkage field. The queue is a simple linked-list structure that can be easily manipulated by driver, but DevHelp services provide an alternative queue management method.

PushReqPacket adds a packet to the end of a queue, and PullReqPacket removes the first packet from the queue. PullParticular removes a packet, identified by its far address, from the queue. SortReqPacket is used by block drivers to enqueue read/write packets in order by sector number (sometimes known as *elevator seeking*), thereby minimizing disk-head movement.

This request queue function group also includes AllocReqPacket and FreeReqPacket, which enable a driver

to allocate and release additional request packets from the system pool. Certain drivers may need to enqueue more than one packet to complete an I/O request, and these additional packets must be allocated from memory managed by the kernel to ensure addressability in either real or protected mode. The allocate function returns a bimodal pointer to a maximum-sized request packet. Because typical versions of OS/2 have only about 32 packets in the pool, they should be carefully allocated. Specifically, request packets should not be used for temporary data storage; memory-management functions should be used instead.

Process management services. While OS/2 drivers are not really processes, they do at times behave like process threads and share the CPU with other threads. However, they cannot use the normal API functions for process management. Instead, a driver must, in the kernel mode or user mode, call the Block function to suspend execution while awaiting an I/O interrupt or timeout. Then the interrupt handler calls the Run function to wake the driver thread up again.

When Block and Run are called, one of the parameters is a 32-bit event identifier. Normally, this is the address of some type of data structure, such as the request packet, that contains information about the I/O operation in progress. If you use this convention, your event identifier likely will be unique. Because another driver could use the same event identifier, you should always use the following loop when blocking: (1) disable interrupts; (2) call Block; (3) disable interrupts; (4) check the appropriate data structures to determine if the awaited event has occurred—if not, return to step 2; (5) enable interrupts.

Block always returns with interrupts enabled, so you must disable them before checking driver data structures that could be changed by an interrupt routine. This loop provides foolproof protection against duplicate event identifiers, as long as there is a unique way to determine from the contents of a data area whether or not the I/O process is complete. If your driver is awakened by another driver using the same identifier, you simply will repeat the Block operation.

A second parameter for Block is a 32-bit timeout value specifying the maximum number of milliseconds to wait before resuming execution. This value should be used whenever you can predict how long an I/O operation ought to take. For example, if you send a character out to a serial port operating at 9600 baud, the port should respond with an interrupt in about 1 ms. So you might set a timeout value of 100 ms to detect a port failure. If you do not want to time the operation, use a timeout value of -1.

Block also accepts a flag indicating if you want the driver to be awakened when the kernel detects some unusual event, such as the death of the process currently using the driver. Normally, you should allow this type of forced resumption. When it occurs, clean up the operation in progress and return an appropriate error code.

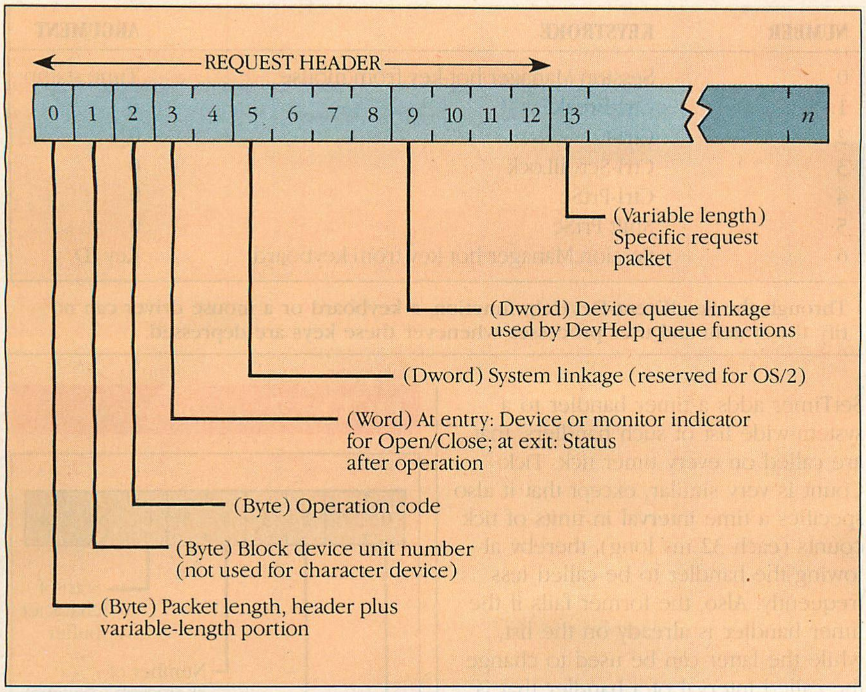
Yield and TCYield are similar to Block, except that the driver is not waiting for an I/O interrupt. Instead, it is notifying the kernel that it is willing to yield the CPU to another thread for a while. Drivers should use these functions when they are involved in some lengthy operation such as a complex computation, a long data transmission via programmed I/O (PIO), or an I/O polling loop. Yield indicates that the driver is willing to give up control to any thread that is ready to run, whereas TCYield yields only to time-critical threads.

In many simple cases, a driver does not need to use any of these process management functions because the kernel automatically blocks the thread if the driver strategy routine returns with the completion flag (bit 8) reset in the request packet status word. Then the interrupt routine need only call DevDone when the operation finishes. DevDone sets the completion bit in the request packet and wakes up any threads that were waiting on that packet. The driver must place any other pertinent status and return information in the packet before calling DevDone.

The final process management service is SendEvent. This function sends an *event*, also known as a *signal*, to OS/2, which then alerts all processes that have registered routines for handling a particular signal. The SendEvent function takes two arguments: the event number and an event argument. Numbers and arguments for the seven events currently defined are listed in table 2. An application process can register handlers only for events 1 and 2 (Ctrl-C and Ctrl-Break, respectively); the others are reserved for use by OS/2.

These events all originate from the keyboard and mouse drivers, so you do not have to use the SendEvent function unless you are planning to replace either of these drivers.

FIGURE 1: *Driver Request Header*



The queue-linkage field points to the next request packet that is waiting on the driver's work-pending queue. The device driver may use either the DevHelp services or its own routines in order to manage the request queue.

Semaphore services. Some drivers communicate with their callers via semaphores. Typically, the caller allocates a system semaphore or a private semaphore (also called a RAM semaphore) via the normal API services and passes it to the kernel as a parameter in an API function requesting I/O services. The kernel then passes it to the driver in the request packet.

The driver must call SemHandle to convert that handle into a form suitable for use in the various driver contexts. Then the driver uses SemRequest to test and set the semaphore and SemClear to release it. When calling SemRequest, the driver can specify one of three actions to take if the semaphore is already owned: wait indefinitely, wait a specified interval and return with an error, or return immediately with an error. When the driver has finished with the semaphore, it calls SemHandle again to release it.

Because private semaphores reside in the user's data space, many special restrictions are placed on their access from drivers. You can tell if the user has passed you the handle of a private semaphore by comparing the handle returned from SemHandle. If it is the same as the user's handle, then it refers to a private semaphore. In that case, its accessibility must be verified by calling VerifyAccess, and its segment or selector must be locked in physical memory via the Lock function.

Character queue services. Many drivers use character queues to buffer the information they receive from or send to a programmed I/O device. The DevHelp services include several functions that simplify the management of these queues. This set of functions is a convenience, not a necessity.

A character queue consists of three words followed by a buffer, as shown in figure 2. This structure must be defined in the driver's data segment. To use these services, the driver first calls QueueInit to initialize this structure, then calls QueueWrite and QueueRead to move characters into and out of the queue. QueueFlush throws away everything in the queue.

Interrupt services. A driver attaches itself to interrupts via the DevHelp functions SetIRQ and SetROMVect. The former is used for hardware interrupts, while the latter is used for software interrupts, especially those associated with the ROM BIOS functions used by DOS applications. UnSetIRQ is used to detach from a hardware interrupt. Detaching from a software interrupt is done by reinstalling the previous handler, whose address is returned by the SetROMVect function.

When the hardware interrupt hooked with SetIRQ occurs, the kernel

saves all registers, sets up CS and DS for the driver's segment and the current addressing mode, and calls the driver's hardware interrupt routine. If the interrupt level is being shared by several drivers, you must first determine if this interrupt is from your device. If not, set the carry flag and return immediately via a RETF instruction. This indicates that you are not claiming the interrupt.

If the interrupt is from your device, perform the necessary actions, and then execute the following return sequence: (1) Disable interrupts; (2) call the EOI function; (3) clear the carry flag; (4) return to the kernel (via RETF, not IRET).

The call to EOI indicates that the driver is finished with its interrupt processing and is ready to accept another interrupt. If you call EOI to release the interrupt controller before completing all interrupt work, then your interrupt routine should be designed carefully so that it can be reentered while that additional work is going on. Failure to do so may result in a nested interrupt loop, which will lock up the system.

When you hook into a software interrupt via SetROMVector, OS/2 plugs your interrupt handler's address directly into the interrupt vector. Therefore, the interrupt handler receives control directly, without intervention of the kernel, so it must itself save registers and load DS so that it points to the driver's data segment. To facilitate this, SetROMVector saves the driver's DS at the specified place in the driver code segment. Also, because entry is via an INT instruction and not a CALL, exit must be via IRET, not RETF.

The SetROMVector call will fail if you attempt to hook into a hardware interrupt. On the IBM systems, the hardware interrupts are 08H to 0FH, 50H to 57H, and 70H to 77H.

Unlike hardware interrupts, software interrupts occur only in real mode and can be preempted by the OS/2 task scheduler—that is, while you are processing a software interrupt, OS/2 can grant control of the CPU to another process with higher priority. You must use ROMCritSect to prevent this from happening during a critical operation, such as an I/O sequence that is time-critical or could leave a device in an awkward state. This is less drastic than disabling interrupts.

Timer services. This group of timer services allows the driver to perform certain processing at regular intervals, for example, to poll noninterrupting devices or to set timeout intervals.

TABLE 2: *Event Numbers and Arguments*

NUMBER	KEYSTROKE	ARGUMENT
0	Session Manager hot-key from mouse	Time stamp
1	Ctrl-Break	0
2	Ctrl-C	0
3	Ctrl-ScrollLock	0
4	Ctrl-PrtSc	0
5	Shift-PrtSc	0
6	Session Manager hot-key from keyboard	Key ID

Through the SendEvent DevHelp function, a keyboard or a mouse driver can notify the system and user processes whenever these keys are depressed.

SetTimer adds a timer handler to a system-wide list of such handlers, that are called on every timer tick. TickCount is very similar, except that it also specifies a time interval in units of tick counts (each 32 ms long), thereby allowing the handler to be called less frequently. Also, the former fails if the timer handler is already on the list, while the latter can be used to change the called interval of a handler that is already on the list. ResetTimer removes a handler from the timer list.

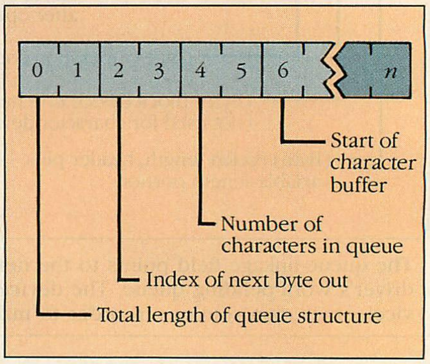
A timer handler is called at the timer interrupt level and must save and restore registers. It should do its work as quickly as possible and then return via a RETF instruction instead of IRET.

The SchedClockAddr function is used only by the system clock driver to obtain the address of the OS/2 clock tick handler. The clock driver attaches itself to a timer hardware interrupt and calls the tick handler on each timer interrupt with AL containing the number of milliseconds since the last call. The tick handler then distributes timing signals throughout the system. It calls at appropriate intervals the handlers registered by SetTimer and TickCount.

Monitor services. Monitors are special processes that examine and possibly modify the data stream flowing between a character device and the processes currently using that device. Monitoring a given device is possible only if the driver has been designed to allow it. This article describes the driver-monitor interface from the driver end (*PC Tech Journal* plans to publish, in the near future, an article on OS/2 monitors that will give a description from the monitor's end).

The first step in providing monitor support is to create an empty monitor chain to which one or more monitors can be linked. The driver does this by calling the DevHelp function known as MonCreate. In the most straightforward approach, the driver creates the empty chain at initialization time. Alternatively, the driver may wait until a process actually registers itself as a monitor.

FIGURE 2: *Data Queue*



If the device driver uses DevHelp services to manage a character queue for buffering characters to or from the device, then it must provide this data structure in its data segment.

The driver calls MonCreate with two addresses. One points to a buffer to receive the output from the last monitor in the chain. The first word of the buffer contains its length in bytes, including the 2 bytes used by the length word. The other address points to a *notification routine* that is called when data are placed in this buffer.

A user process hooks into the monitor chain by a two-step process. First, it calls the OS/2 API function DosMonOpen, specifying in a parameter the device name that it wants to monitor. The kernel passes this along to the driver for that device as an Open command. To distinguish this request packet from one produced by the DosOpen API function, the low-order byte of the status word (see figure 1) is nonzero. The driver may be written in order to create the monitor chain at this point.

Next, the user process calls DosMonReg, passing as parameters the addresses of input and output buffers

and a flag specifying whether the monitor wants to be first in the chain, last in the chain, or does not care. The buffers each must be at least 20 bytes larger than the buffer used by the driver in the MonCreate call, and the first word must contain the total length in bytes.

The kernel notifies the driver of the register request by issuing an IOCTL (I/O control) call, passing as data the buffer addresses and the position flag from the user process. If the driver had not previously called MonCreate, it must do so now, and then call the DevHelp function Register, passing it the parameters from the IOCTL call. If more than one monitor requests placement at the head or tail of the chain, only the first one to be registered is so placed.

Once the monitor chain is created and monitors are registered, the driver can send data through the monitor chain. It does this by building a data record for each packet to be processed by the monitors, and calling the DevHelp function MonWrite. The first word in the data record is a flag word containing a byte of system-defined flags and a byte of driver-specific flags. The system flags specify if the data packet is a normal packet, a request to open or close the device, or a request to flush data from the monitor queue.

The remainder of the record contains the actual data. The format of these data, and whether characters are passed singly or in groups, varies from driver to driver. Writing a monitor requires that this information be available in the driver's published specification.

The interface between drivers and monitor processes is controlled by an OS/2 component called the *monitor dispatcher*. At a call to MonWrite, the dispatcher waits until the input buffer of the first monitor in the chain is idle, then it copies the driver's data record into that buffer and marks it busy.

When the monitor calls the API function DosMonRead, the system copies the record from the input buffer to a specified local buffer and marks the input buffer idle. The monitor process can absorb the record, in which case it disappears from the device input stream; or the process can call DosMonWrite to place the same record, a modified version of the record, or several new records back into the chain. DosMonWrite copies each of these records from the local buffer into the monitor's output buffer, and then it alerts the monitor dispatcher.

The dispatcher waits until the next monitor's input buffer is idle and then repeats this procedure with that monitor and with each succeeding one in the chain. When the last monitor process calls DosMonWrite, the monitor dispatcher copies the final data record into the driver's monitor buffer and calls the notification routine. The dispatcher received the addresses of the final buffer and of the notification routine with the MonCreate call. At that point, the driver can process the data as appropriate, sending them to the caller or to the output device, depending on the direction of the transfer.

If the monitor chain is empty, then the dispatcher immediately puts any data records passed into it by MonWrite into the driver's output buffer, and calls the notification routine. Therefore, if the driver creates the chain unconditionally, it can process data identically without testing whether monitors have in fact been registered.

This is an elegant scheme for hooking into device data streams, which is essential for the so-called pop-up utilities, such as Borland's SideKick and RoseSoft's ProKey, that

TABLE 3: *Information Returned by GetDOSVar*

AX	RETURNS POINTER TO
1	Bimodal segment address of the global information area (see table 4).
2	Virtual address of the local information area (see table 5). The pointer is not valid during interrupts.
3	Reserved.
4	Stand-alone dump facility (a debugging aid).
5	OS/2 warm boot routine.
6	Reserved.
7	Yield flag (set whenever a higher-priority thread is ready to run): the driver can test this byte to see if Yield should be called.
8	Time-critical yield flag (set when a time-critical thread is ready to run): the driver can test this byte to see whether TCYield needs to be called.
9	Reserved.
10	Reserved.
11	Virtual address of code page tag (used for national language support) for the current process (pointer not valid during interrupts).

The GetDOSVar function returns a pointer to a data area, procedure, or flag. The driver can use it to get information on the state of the system or the calling task.

have become so popular under DOS. It allows very low-level monitoring of devices without disrupting the multitasking and protection features of OS/2. However, monitor processes, especially those for the keyboard, mouse, and screen, must be carefully designed to prevent the entire system from appearing sluggish and unstable.

System services. The most generally useful system services function is GetDOSVar; it returns a bimodal pointer to one of several system variables, depending on the parameter supplied in AX (see table 3). Some of the pointers are to fairly extensive data structures, as shown in tables 4 and 5. All of these variables are read-only.

The remaining DevHelp functions in this group, available only in IBM's version of OS/2, provide convenient driver access to the Advanced BIOS (ABIOS) used on IBM PS/2 computers that allows access to all 16MB of address space and can be used in a multitasking operating system.

WORKING ON THE INSIDE

How are drivers actually constructed? Consider for a moment the driver for the device commonly known as the LPT1, which is a printer attached to the first parallel I/O channel. To all intents and purposes this is a garden-variety device—yet, its driver must support most of the important features, including monitors, hardware interrupts, and BIOS simulation.

The first step in any driver-design effort is to study the hardware specifications. The parallel channel contains only three I/O ports. The first, port 0, is the data register into which the driver writes characters being sent to the printer. Port 1 is the control register containing the five control bits shown in figure 3. Port 2 is a status register with the five status bits as shown in figure 4.

To print a character, the driver writes the output byte to the data port, reads the status port to determine if the printer is not busy, and if not, sets and then resets the *strobe bit* in the control port. The pulsing of the strobe bit causes the printer to accept the byte from the data port; during this transfer the printer sets the *busy bit* in the status port. When the printer completes the transfer and is ready to receive the next character, it resets the busy bit and pulses the *acknowledge bit*. When the *interrupt enable bit* in the control port is set and the channel is assigned to an interrupt level, a hardware interrupt occurs. The interval between the strobe and the acknowledgement may be negligible if the character is transferred into the printer's internal buffer, but may be significant if the printer has to wait for the print head actually to output a character to paper.

The I/O addresses of the parallel channels are placed in a three-word table by the power-on self-test (POST) routine. This table is at location 0040:0008 in the BIOS data area, which the driver can address directly via a bimodal pointer. Machines conforming to the IBM conventions can have three parallel ports with the following I/O address and interrupt level assign-

ments: 3BCH, interrupt level 7; 378H, interrupt level 5; 278H, no assigned interrupt level. These addresses are for port 0; the other two ports are at successively higher I/O addresses.

The POST routine checks the parallel channels in the order listed, and for each one that it finds, it places the I/O address at the next location in the BIOS table. Unused entries in the table contain zero. By DOS and OS/2 conventions, the first table entry is LPT1, the second is LPT2, and so on. Note that the third channel has no provision for interrupts, and so it cannot be handled by this sample driver.

The next step is to decide on the structure of the driver—that is, on your objectives and how you are going to achieve them within the general OS/2 driver-design requirements. The goals for this example driver are to:

- Handle all output that is directed to LPT1 or its synonym, PRN.
- Allow any number of processes to write to LPT1 simultaneously, with the assumption that a spooler will be attached as a monitor to separate the various output streams.
- Provide reasonably good error protection from common printer situations such as "no paper," "jam," or "off-line."

The actual printer driver supplied with OS/2 is considerably more complex than this example, because it can handle all three printer channels, has more extensive support for monitors, supports the complete set of IOCTL operations, and supports code page and font switching.

The example driver handles only LPT1, which is also known as PRN. It does not include code page/font support, and it provides only the minimum set of IOCTL operations. Support for monitors, including a spooler, is fairly complete but quite simple, and a "real" driver would need some tuning for performance and error recovery. Nonetheless, this example driver could replace the standard LPT1 driver.

Although the driver is meant to operate with a spooler as a monitor, it is fully operational even with no spooler attached. When only one process is using the printer, the output appears normally. If several processes print simultaneously, their output will be interspersed, but the operation of the processes will not be impaired.

To keep the example simple, error recovery is somewhat rudimentary. If the printer goes off-line for any reason, the driver polls it indefinitely at half-second intervals until the printer comes back on line or the driver receives a request to flush the printer output queue. That way, if the printer can be restarted, it will resume printing with no loss of data. Unlike in DOS, an error message cannot be returned to the process sending the data, because sending data to the driver and to the printer are asynchronous activities. The process from which the data originated may not even be running at the time the printer error occurs. The alternative would be to pop up an error message, as is done by the OS/2 diskette driver, in order to alert the operator to rectify the situation.

TABLE 4: ***Global Information Segment***

OFFSET, BYTES	LENGTH, BYTES	DESCRIPTION
0	4	Time from 1-1-1970 in seconds
4	4	Current time: milliseconds (ms)
8	1	hours
9	1	minutes
10	1	seconds
11	1	hundredths of seconds
12	2	Time-zone correction in minutes
14	2	Timer interval: 0.1 ms
16	1	Current date: day
17	1	month
18	2	year
20	1	day of week (0 = Sunday)
21	1	OS major version number
22	1	OS minor version number
23	1	OS revision letter
24	1	Current foreground session
25	1	Maximum number of sessions
26	1	Shift count for huge segments
27	1	Protect-only mode (0 = DOS box exists)
28	2	PID of current foreground process
30	1	Dynamic priority (0 = disabled)
31	1	Maximum wait for CPU in seconds
32	2	Minimum timeslice (ms)
34	2	Maximum timeslice (ms)
36	2	Boot drive ID (1 = A:)
38	2	Trace flags

A GetDOSVar call with AX = 1 returns a pointer to the address of this data structure containing system-wide information. The address is bimodal, so the structure is accessible in both protected and real modes and in any driver context.

DRIVER CODE

The code of the example driver is shown in OS2LPT.ASM (listing 1) The various routines are examined in logical order (that is, in order in which they are usually executed), which is not the same order in which they actually appear in the listing.

Assembly parameters. The driver begins with some assembly parameters defining the DevHelp function codes, the layouts of various data structures such as the request packets, and some miscellaneous items. The MONPKT structure defines a monitor data record of 132 bytes, with 4 bytes of header information and 128 bytes of data. The first 2 bytes are the flags, which always begin a monitor data record. The third and fourth bytes hold the identifier of the user process sending the data to the printer; this is provided so that the spooler can create separate spooling files for each process. This is an example of a driver-specific data structure whose layout must be communicated to anyone who is designing a monitor for this device driver.

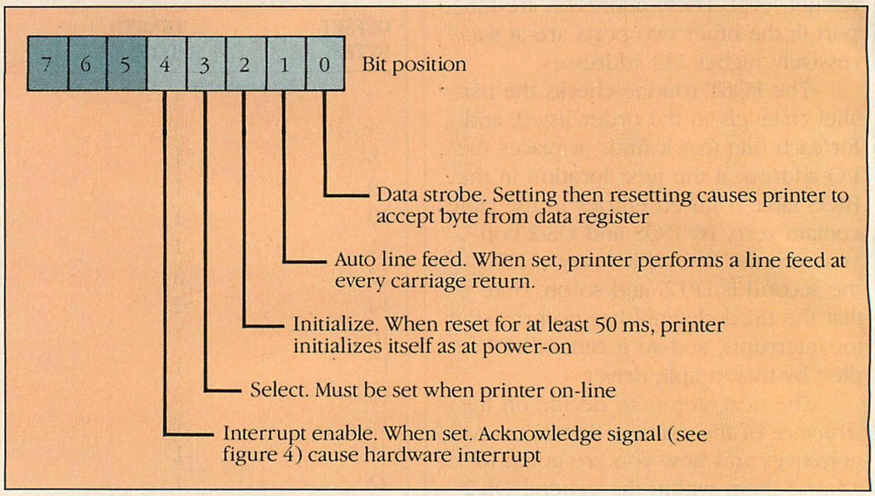
Data segment. The driver data segment begins with the device header blocks for LPT1 and PRN. Next comes a transfer vector table used by the strategy routine to call the functions that process the various request packets. This is followed by two monitor buffers declared with the MONPKT structure: one for sending messages into the monitor chain with MonWrite and one for receiving them from the end of the

TABLE 5: *Local Segment*

OFFSET, BYTES	LENGTH, BYTES	DESCRIPTION
0	2	Current process ID
2	2	Parent process ID
4	1	Current thread priority level
5	1	Current thread priority class
6	2	Current thread ID
8	2	Current session number
10	2	Reserved
12	2	Current foreground process ID

The local information segment identifies the task that requested the system service resulting in the call to the driver. It is only available at task time, because other calls to the driver are asynchronous to any task.

FIGURE 3: *Printer Control Register*



The driver writes to this register to control the operation of the printer, such as initializing and setting polled or interrupt-driven handshaking. Pulsing the strobe bit high then low causes the printer to accept the byte from the data port, provided that the printer is not busy or in an error state.

chain. The receive buffer is 2 bytes longer because it begins with a length word, whereas when sending, the length is passed in a register.

Strategy routine. The strategy routine at label S_LPT in the code segment is the common entry point for all driver requests initiated by user processes. This routine saves the request packet pointer on the stack and sets up the BP register as a stack frame pointer. Then it calls an internal procedure to handle the command code that is located in the request packet.

Initialize. The initialization routine at label S_INIT is at the end of the code segment so that it can be deleted after running once. It first removes the DevHelp address from the request packet, then it loads the resident code and data sizes into the packet.

Next, it checks if the first printer channel is assigned to a device capable of generating interrupts. If not, an error code is returned. Otherwise, the channel address and interrupt level number are saved, and the channel is initialized. The printer is initialized by writing a 1 bit to the control port for at least 50 microseconds, then writing a 0 bit. The time delay is produced by a spin loop with a loop count that should be adequate well beyond 20-MHz 80386 machines. Such loops are often used in drivers, because the OS/2 timing facilities cannot resolve down to such short intervals. However, this approach often forces driver changes as systems get faster.

The final initialization steps are to create a monitor chain and attach the driver to the hardware and software (BIOS) interrupts.

Uninstall. The Uninstall procedure at label S_RMV is called if the OS/2 driver loader decides that the driver should be removed. It simply deletes the monitor chain and detaches from the interrupts. OS/2 guarantees to call this routine before any of the I/O traffic has started, so the monitor chain should be empty and no printer interrupts should be pending.

Write, notify, interrupt. The write procedure at label S_WRT gains control when a user process wants to send data to the printer. The write routine, together with the notify and interrupt routines (located at labels **NOTIFY** and **I_LPT**, respectively), are the heart of the device driver.

The user process can send nearly 64KB of data in one request, but the monitor chain can absorb only 128 bytes at a time because that is the size of the monitor-chain buffer. Therefore, the code sequence beginning at label **WRT1** breaks the user's data into several monitor packets if necessary.

Each message is sent to the monitor chain via MonWrite with the automatic-synchronization option. MonWrite will not return until the message has been absorbed.

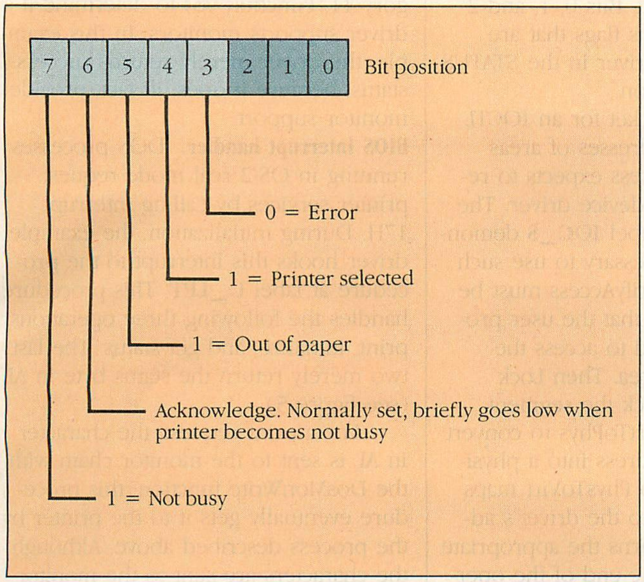
The address of the user's data buffer, passed by the kernel in the request packet, is in physical format. The driver must map it into virtual form with PhysToVirt before *each* call to MonWrite, because a previous mapping may become invalid if the driver blocks waiting for MonWrite to return.

The monitor dispatcher passes the data through the chain of monitors. If one of the monitors is a spooler, it collects the data records into separate files by process ID and passes them down the chain in a different order than received. Other monitors might, for example, translate printer control codes from one make of printer to another. Eventually, the dispatcher receives a data record from the last monitor, places in the driver's monitor buffer (label **MON** in the data segment) and calls the **NOTIFY** routine, which sends the data to the printer.

In this example, **NOTIFY** can receive four record types from the monitor chain: open, close, flush, and write. Open and close requests are ignored because they have been completely processed by the time they traverse the chain. The receipt of a flush record from the monitor chain indicates that the flush process has been completed, so the flush-in-progress bit in the driver status word is reset. Each write message is processed as follows:

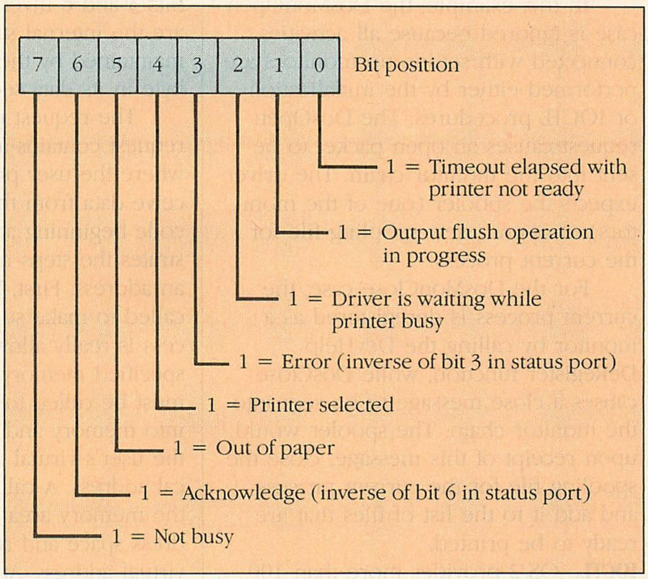
1. Check the printer status. If it is ready, send the next character and continue to repeat this step until all of the data are sent. Then return to the monitor dispatcher.
2. If the printer is not ready and no error indicators are set, then it is probably busy printing. Enable the printer interrupt, turn on the waiting bit in the status word, and block until restarted by the interrupt routine (indicating that the printer is ready again) or until the specified timeout period elapses. In the

FIGURE 4: ***Printer Status Register***



The driver may poll the printer by repeatedly reading this register, or it may wait for an interrupt caused by a pulse on the acknowledge bit. The printer resets and sets this bit after it accepts one character and is ready for the next.

FIGURE 5: ***Printer Status Byte***



The driver returns this byte to its caller in response to a status request. Five of the bits are copies or inverses of bits from the Printer Status Register; the other three bits indicate conditions within the driver software.

former case, you must continue sending data at step 1.

3. If an error indicator is set or a timeout occurs in the previous step, begin a timed polling loop to check the printer status every 500 ms until it becomes ready again or until the flush bit is set in the driver status word (label **STATUS** in the data segment). In the former case, resume the data transmission process at step 1 above. In the case of the flush bit, the monitor is in the process of flushing its data, so abandon any data in the output buffer, disable the printer interrupt, and return to the monitor dispatcher. If neither of these conditions occurs, continue polling indefinitely.

The interrupt routine at label **I_LPT** is normally entered after the printer finishes processing the data sent by the notification routine. **I_LPT** first tests the wait bit in the status word to determine whether the notification routine is waiting for the printer to become available. If that flag is set, **I_LPT** resets it and unblocks the notification routine by calling the DevHelp Run function. In any case, **I_LPT** calls EOI to reset the interrupt controller and returns to the system.

The notification and interrupt routines use the address of the printer port table in the BIOS data area as the event identifier for the Block/Run sequence. In addition, when **NOTIFY** wakes up, it tests the wait flag for further confirmation that it was indeed the printer interrupt routine that woke it, not some other process that just happened to use the same event ID. If the wait flag is still set, **NOTIFY** then goes back to sleep.

There are many other ways to handle printer I/O than the approach implemented here. For example, **NOTIFY** could remove the data from the monitor buffer and place them into a queue. If the device were idle, **NOTIFY** would then send the first character to it, and let the interrupt routine send the remaining characters. This is the appropriate method for slower printers without internal buffers. However, modern dot-matrix and laser printers have large internal buffers that receive data quite rapidly; therefore, it is best to avoid the high overhead that would occur if only one character were sent on each interrupt. Also, filling the printer buffer is not recommended at the interrupt level, because this procedure could take hundreds of milliseconds, and interrupt routines should be extremely fast for good OS/2 performance.

Output status. The routine at label **S_STAT** handles the "get output status" request to the device driver. It always returns "success" to indicate that the driver is ready to accept output characters, even if the printer is actually offline. This is a reasonable approach if a spooler monitor is running, because all output data are buffered to a file, regardless of the state of the printer.

Output flush. The routine at label **S_FLUSH** is called when the driver gets a command to flush its output queue. It sets the flush bit in the status word to indicate that a flush is in progress, then calls MonFlush. This DevHelp function sends into the monitor chain a data packet containing only a flag word identifying the packet as a flush request. It is up to each monitor to clean up its internal buffers and otherwise dispose of any data it might be holding. Each monitor *must* pass along the packet, because the dispatcher will not accept any MonWrite requests until the flush packet reaches the notification routine. At that time, the monitor chain is completely empty, which means that all output data are either saved in spool files or sent to the printer. When the **NOTIFY** routine receives this packet, it merely turns off the flush bit to indicate the completion of the operation.

Setting the flush flag also causes **NOTIFY** to abort a printer data transmission operation or not-ready polling loop. Thus, any data still in the monitor chain are discarded as they reach the **NOTIFY** routine.

Open, close. The strategy routine receives an open request packet when a user process calls DosOpen or a monitor process calls DosMonOpen. Similarly, a close request packet can be the result of calling either DosClose or DosMonClose. A request for opening or closing of a monitor is distinguished

by a nonzero value in the low-order byte of the status word.

In this example, the DosMonOpen case is ignored because all activities connected with setting up monitors are performed either by the initialization or IOCTL procedures. The DosOpen request causes an open packet to be sent into the monitor chain. The driver expects the spooler (one of the monitors) to open a new spooling file for the current process.

For the DosMonClose case, the current process is deregistered as a monitor by calling the DevHelp DeRegister function, while DosClose causes a close message to be sent into the monitor chain. The spooler would, upon receipt of this message, close the spooling file for the current process and add it to the list of files that are ready to be printed.

IOCTL. OS/2 provides more than 100 IOCTL functions, divided into 11 categories. The example driver supports only the following three of these functions: Get Printer Status, Register Monitor, and Query Monitor Support. The rest of the IOCTL functions are treated as invalid commands.

Get Printer Status (category 5, function 101) returns a status byte (see figure 5) into the user's data area. Bits 3 through 7 are read directly from the status port of the parallel channel, with bits 3 and 6 inverted. Bits 0, 1, and 2 are the internal status flags that are maintained by the driver in the STATUS byte in its data section.

The request packet for an IOCTL request contains addresses of areas where the user process expects to receive data from the device driver. The code beginning at label IOC_S demonstrates the steps necessary to use such an address. First, VerifyAccess must be called to make sure that the user process is really allowed to access the specified memory area. Then Lock must be called to lock the segment into memory and VirtToPhys to convert the user's virtual address into a physical address. A call to PhysToVirt maps the memory area into the driver's address space and returns the appropriate virtual address. At the end of the operation, Unlock removes the memory lock, and UnPhysToVirt restores the previous addressing mode and releases the mapping resources.

The OS/2 kernel issues the IOCTL function Register Monitor (category 10, function 64) in response to a DosMonReg call. The driver calls the DevHelp Register function, passing it the data from the IOCTL packet.

A user process uses the IOCTL function Query Monitor Support (Category 11, Function 96) to determine if a driver supports monitors. In this example, the driver merely returns success status, because it does, in fact, provide monitor support.

BIOS interrupt handler. DOS processes running in OS/2 real mode request printer services by calling interrupt 17H. During initialization, the example driver hooks this interrupt to the procedure at label U_LPT. This procedure handles the following three operations: print, initialize, and get status. The last two merely return the status byte in AL (see figure 5).

For a print request, the character in AL is sent to the monitor chain with the DosMonWrite function; this procedure eventually gets it to the printer by the process described above. Although the characters are sent to the monitor chain one by one, one of the monitors might buffer them into 128-character chunks before passing them back to the NOTIFY procedure for printing. The print function also returns the status byte in the AL register.

Because the user interrupt handler gains control directly from a software interrupt with no intervention from the kernel, it must save and restore registers and establish addressability to its data segment. Also, it must use an IRET instruction, not RETF, to exit.

Designing drivers in a new operating environment can be a perplexing task. However, the operating system provides DevHelp services to support the developer's implementation burden. In this article we have examined components such as initialization routines, strategy routines, interrupt routines, and monitor support. The device driver has been written all the way down to the hardware level where it communicates with the I/O ports.

The development of OS/2 drivers is quite complex. The OS/2 technical documentation contains a wealth of additional information. Also, for $350, Microsoft provides an OS/2 Device Driver Development Kit that includes some special driver debugging tools. Although system-level programming for OS/2 is undoubtedly more difficult than for DOS, the trade-off is that applications development should be simpler, because the operating system can handle many more details.

David A. Schmitt is president of Lattice, Inc., now a subsidiary of SAS Institute. Mr. Schmitt has recently directed the adaptation of the Lattice C library to OS/2.

LISTING 1: OS2LPT.ASM

```
        PAGE    60,132
        TITLE   Line Printer Driver
        NAME    LPT
        .286
;*******************************************************************
; Miscellaneous assembly parameters
IOSEG   EQU     40H             ; BIOS data area segment address
IOTAB   EQU     8               ; offset to printer table in BIOS area
IOLPT1  EQU     3BCH            ; normal LPT1 I/O address
IOLPT2  EQU     378H            ; normal LPT2 I/O address
IOINT1  EQU     7               ; normal LPT1 interrupt level
IOINT2  EQU     5               ; normal LPT2 interrupt level
TIMEOUT EQU     10000           ; timeout in milliseconds, -1 for no timeout
;*******************************************************************
;
; Device Helper Service information
DevDone         EQU     1
Block           EQU     4
Run             EQU     5
LockSeg         EQU     13H
UnlockSeg       EQU     14H
PhysToVirt      EQU     15H
VirtToPhys      EQU     16H
SetROMVector    EQU     1AH
SetIRQ          EQU     1BH
UnSetIRQ        EQU     1CH
MonCreate       EQU     1FH
Register        EQU     20H
DeRegister      EQU     21H
MonWrite        EQU     22H
MonFlush        EQU     23H
GetDOSVar       EQU     24H
ROMCritSect     EQU     26H
VerifyAccess    EQU     27H
EOI             EQU     31H
UnPhysToVirt    EQU     32H
;*******************************************************************
; Driver request packet data structure
;*******************************************************************
DRVPKT  STRUC
P_LEN   DB      ?               ; packet length
P_UNIT  DB      ?               ; unit number
P_CMD   DB      ?               ; command (operation) code
P_STAT  DW      ?               ; status code
P_SYSL  DD      ?               ; system queue linkage
P_DEVL  DD      ?               ; device queue linkage
DRVPKT  ENDS
;
; Flags in P_STAT
PS_ERR  EQU     8000H           ; error flag
PS_ERRD EQU     4000H           ; device error flag
PS_BUSY EQU     0200H           ; busy flag
PS_DONE EQU     0100H           ; completion flag
;
; Error codes in P_STAT
;
PE_CMD  EQU     5               ; bad command code
PE_OUT  EQU     9               ; out of paper
PE_WRT  EQU     0AH             ; write error
PE_GEN  EQU     0CH             ; general failure
;
; Request packet extension for initialize operations (code 0)
;
PKTI    STRUC
        DB      13 DUP(?)       ; header
        DB      ?               ; reserved
P_HLP0  DW      ?               ; DevHlp address, returns CS size
P_HLP1  DW      ?               ; DevHlp address, returns DS size
PKTI    ENDS
;
; Request packet extension for write operation (codes 08H and 09H)
;
PKTW    STRUC
        DB      13 DUP(?)       ; header
        DB      ?               ; unused
P_BADD  DD      ?               ; buffer address
P_BLEN  DW      ?               ; buffer length
PKTW    ENDS
;
; Request packet extension for IOCTL operation
;
PKTIOC  STRUC
        DB      13 DUP(?)       ; header
P_CAT   DB      ?               ; category
P_FUN   DB      ?               ; function
P_PARM  DD      ?               ; parameter area pointer
P_DATA  DD      ?               ; data area pointer
PKTIOC  ENDS
;
; Data area for IOCTL "register" function
;
IOCREG  STRUC
R_POS   DB      ?               ; position code
R_NDX   DW      ?               ; index
R_IN    DD      ?               ; input buffer address
R_OUT   DW      ?               ; output buffer offset
IOCREG  ENDS
;*******************************************************************
;
; Monitor chain buffer structure
;
MONLEN  EQU     132             ; size of monitor buffer
MONHDR  EQU     4               ; size of monitor buffer header
MONPKT  STRUC
M_FLAG  DW      0               ; flags
M_PID   DW      0               ; user process id
M_DATA  DB      MONLEN-MONHDR DUP(?)    ; data area
MONPKT  ENDS
;
; Monitor flags (byte 0 of M_FLAG)
;
MF_OPEN EQU     1               ; open flag
MF_CLOS EQU     2               ; close flag
MF_FLSH EQU     4               ; flush flag
        SUBTTL  Data Segment
        PAGE
;*******************************************************************
; Define the data segment.  The name _DATA is used here for compatibility
; with most OS/2 C compilers, but any name is acceptable.
;*******************************************************************
_DATA   SEGMENT WORD PUBLIC 'DATA'
;*******************************************************************
;
; This is the LPT1 device driver header.  No other data can precede it.
;
;*******************************************************************
LPT1    DW      PRN,SEG PRN     ; linkage
        DW      88C0H           ; device type bits...
                                ;  15    => character device
                                ;  11    => open/close support
                                ;  09-07 => driver type 1 (OS/2)
                                ;  06    => IOCTL supported
        DW      S_LPT           ; offset to strategy routine
        DW      -1              ; reserved
        DB      'LPT1    '      ; device name
        DW      4 DUP(0)        ; reserved
;*******************************************************************
; This is the PRN device driver header.  LPT1 and PRN are synonyms.
;*******************************************************************
PRN     DD      -1              ; linkage (-1 indicates end of header list)
        DW      88C0H           ; device type bits...
                                ;  15    => character device
                                ;  11    => open/close support
                                ;  09-07 => driver type 1 (OS/2)
                                ;  06    => IOCTL supported
        DW      S_LPT           ; offset to strategy routine
        DW      -1              ; reserved
        DB      'PRN     '      ; device name
        DW      4 DUP(0)        ; reserved
;*******************************************************************
; Driver command table
CMDTBS  DW      S_INIT          ; 00 => Initialize
        DW      CMDBAD          ; 01 => Check media
        DW      CMDBAD          ; 02 => Build BPB
        DW      CMDBAD          ; 03 => Reserved
        DW      CMDBAD          ; 04 => Read
        DW      CMDBAD          ; 05 => Peek
        DW      CMDBAD          ; 06 => Get input status
        DW      CMDBAD          ; 07 => Flush input buffer
```

```
        DW      S_WRT           ; 08 => Write
        DW      S_WRT           ; 09 => Write and verify
        DW      S_STAT          ; 0A => Get output status
        DW      S_FLUSH         ; 0B => Flush output buffer
        DW      CMDBAD          ; 0C => Reserved
        DW      S_OPEN          ; 0D => Open
        DW      S_CLOSE         ; 0E => Close
        DW      CMDBAD          ; 0F => Check removable media
        DW      S_IOCTL         ; 10 => I/O control
        DW      CMDBAD          ; 11 => Reset media
        DW      CMDBAD          ; 12 => Get logical drive map
        DW      CMDBAD          ; 13 => Set logical drive map
CMDTBE  DW      S_RMV           ; 14 => Remove driver (de-install)
CMDMAX  EQU     (CMDTBE-CMDTBS)/2
;*****************************************************************************
; Monitor buffer
MON     DW      MONLEN          ; receive buffer, used by notify routine
MONR    MONPKT  <>
MONS    MONPKT  <>              ; send buffer, used by strategy routine
;*****************************************************************************
; Driver status flags
STATUS  DB      0               ; driver status flags
SF_TIME EQU     1               ; set if timer is running
SF_FLSH EQU     2               ; set if flushing
SF_WAIT EQU     4               ; set if waiting for interrupt
;*****************************************************************************
; Miscellaneous data items
;
DEVHLP  DD      0               ; DevHlp function address
IOTADD  DW      IOTAB,IOSEG     ; points to printer table in BIOS RAM
IOLPT   DW      0               ; current printer I/O address
IOINT   DW      0               ; current printer interrupt level
IOTIME  DD      TIMEOUT         ; timeout value for printer delay
MONHAN  DW      0               ; monitor handle
;*****************************************************************************
;
; End of data segment
;
END_DS  EQU     $               ; used by S_INIT
_DATA   ENDS
        SUBTTL  Code Segment
        PAGE
;*****************************************************************************
;
; Define the code segment.  The name _TEXT is used for compatibility with
; most OS/2 C compilers, but any name is acceptable.  The ASSUME statement
; indicates what values the driver expects in the segment registers.
;
;*****************************************************************************
_TEXT   SEGMENT WORD PUBLIC 'CODE'
        ASSUME  CS:_TEXT,DS:_DATA
;*****************************************************************************
;
; This is the strategy routine.  It is entered via a far call, with the
; request packet pointer in ES:BX.  The packet pointer is valid in both
; real and protected mode.
;
;*****************************************************************************
S_LPT   PROC    FAR
        PUSH    ES              ; save request packet pointer
        PUSH    BX
        MOV     BP,SP           ; set up stack frame pointer
        XOR     AX,AX           ; get driver command code
        MOV     AL,ES:[BX].P_CMD
        CMP     AL,CMDMAX       ; check if in range
        JA      CMDBAD          ; error if not
        MOV     DI,AX           ; call the command routine
        ADD     DI,DI
        CALL    CMDTBS[DI]
        POP     BX              ; restore packet pointer
        POP     ES
        RETF                    ; return to caller
;*****************************************************************************
; The following labels are convenient exit points for strategy routines.
; SS:SP must point to the return address, and the request packet pointer
; must be just above the return address.
;*****************************************************************************
;
; Come here for bad commands
;
CMDBAD: MOV     AX,PE_CMD+PS_ERR
;
; Common exit point, with status code in AX
;
DONE:   MOV     BP,SP           ; set frame pointer
        LES     BX,[BP+2]       ; get packet pointer
        OR      AX,PS_DONE      ; set completion flag
        MOV     ES:[BX].P_STAT,AX ; store codes
        RETN
;
; Come here for general errors
;
ERRGEN: MOV     AX,PE_GEN+PS_ERR
        JMP     DONE
;*****************************************************************************
; S_WRT -- Initiate a write operation
;
S_WRT:  TEST    STATUS,SF_FLSH
        JNZ     RETOK                   ; ignore write request if flushing
        CALL    GETPID                  ; get user process id
        JC      ERRGEN                  ; branch if error
        PUSH    AX                      ; save it on stack
        PUSH    0                       ; clear user buffer index
        PUSH    ES:[BX].P_BLEN          ; copy user buffer length/address
        PUSH    WORD PTR ES:[BX].P_BADD+2
        PUSH    WORD PTR ES:[BX].P_BADD
        MOV     BP,SP                   ; BP is now frame pointer
;
; Call MonWrite one or more times to process the user's data
;
WRT1:   MOV     CX,[BP+4]               ; get residual length
        CMP     CX,MONLEN-MONHDR
        JNA     WRT2                    ; branch if fits in monitor buffer
        MOV     CX,MONLEN-MONHDR        ; else process part of the data
WRT2:   MOV     AX,[BP]                 ; get user buffer address
        MOV     BX,[BP+2]
        PUSH    CX                      ; save current length
        MOV     DX,0100H+PhysToVirt     ; convert address into ES:DI
        CALL    DEVHLP
        POP     CX                      ; restore current length
        JC      WRT4                    ; branch if error
        SUB     [BP+4],CX               ; reduce residual length
        ADD     DI,[BP+6]               ; compute next byte address
        ADD     [BP+6],CX               ; update buffer index
        LEA     SI,MONS.M_DATA          ; move data to monitor send buffer
        PUSH    CX
WRT3:   MOV     AL,[SI]
        MOV     ES:[DI],AL
        INC     SI
        INC     DI
        LOOP    WRT3
        POP     CX
        ADD     CX,MONHDR               ; adjust length to include header
        MOV     AX,[BP+8]               ; move process id to header
        MOV     MONS.M_PID,AX
        MOV     MONS.M_FLAG,0           ; reset all flags
        LEA     SI,MONS                 ; set pointer to data record
        MOV     AX,MONHAN               ; get monitor handle
        MOV     DX,MonWrite             ; DH=0 to synchronize in dispatcher
        CALL    DEVHLP                  ; perform monitor write
        JC      WRT4                    ; branch if error
        CMP     WORD PTR [BP+4],0
        JNE     WRT1                    ; loop till residual length is zero
        MOV     DL,UnPhysToVirt         ; restore prior addressing mode
        CALL    DEVHLP
        ADD     SP,8                    ; discard local stack frame
;
; Come here when operation has completed successfully
;
RETOK:  XOR     AX,AX                   ; clear error code
        JMP     DONE                    ; return
;
; Come here for write error
;
WRT4:   MOV     AX,PE_WRT+PS_ERR        ; load error code
        ADD     SP,8                    ; discard local stack frame
        JMP     DONE                    ; return
;*****************************************************************************
;
; S_STAT -- Get output status
;
S_STAT: JMP     RETOK                   ; return "not busy"
```

```
;***********************************************************************
;
; S_FLUSH -- Flush output queue
;
S_FLUSH: OR     STATUS,SF_FLSH          ; set flush flag
         MOV    AX,MONHAN               ; send flush message to monitors
         MOV    DL,MonFlush
         CALL   DEVHLP
         JNC    RETOK                   ; exit if successful
         JMP    ERRGEN                  ; else signal general failure
;***********************************************************************
;
; S_OPEN -- Open the printer
;
S_OPEN: CMP     BYTE PTR ES:[BX].P_STAT,0
         JNE    OPEN1                   ; ignore if DosMonOpen
;
; Come here to handle DosOpen
;
         MOV    MONS.M_FLAG,MF_OPEN     ; set open flag in header
         CALL   GETPID                  ; put process id into header
         JC     CLSO
         MOV    MONS.M_PID,AX
         MOV    CX,MONHDR               ; set message length
         LEA    SI,MONS                 ; set pointer to message
         MOV    DX,MonWrite             ; DH=0 to synchronize in dispatcher
         CALL   DEVHLP                  ; send the message
         JC     CLSO
OPEN1:   JMP    RETOK
;***********************************************************************
;
; S_CLOSE -- Close the printer
;
S_CLOSE: CMP    BYTE PTR ES:[BX].P_STAT,0
         JNE    CLS1                    ; branch if DosMonClose
;
; Come here to handle DosClose
;
         MOV    MONS.M_FLAG,MF_CLOS     ; set close flag in header
         CALL   GETPID                  ; put process id into header
         JC     CLSO
         MOV    MONS.M_PID,AX
         MOV    CX,MONHDR               ; set message length
         LEA    SI,MONS                 ; set pointer to message
         MOV    DX,MonWrite             ; DH=0 to synchronize in dispatcher
         CALL   DEVHLP                  ; send the message
         JNC    RETOK
CLSO:    JMP    ERRGEN
;
; Come here to handle DosMonClose
;
CLS1:    CALL   GETPID                  ; get monitor process id
         JC     CLSO
         MOV    BX,AX
         MOV    AX,MONHAN               ; de-register this monitor
         MOV    DL,DeRegister
         CALL   DEVHLP
         JC     CLSO                    ; branch if failure
         JMP    RETOK
;***********************************************************************
;
; S_IOCTL -- I/O control operation
;
S_IOCTL: CMP    ES:[BX].P_CAT,5
         JNE    IOCO
         CMP    ES:[BX].P_FUN,65H
         JE     IOC_S                   ; branch if IOCTL 5.101, STATUS
IOCO:    CMP    ES:[BX].P_CAT,10
         JNE    IOC1
         CMP    ES:[BX].P_FUN,40H
         JE     IOC_R                   ; branch if IOCTL 10.64, REGISTER
IOC1:    CMP    ES:[BX].P_CAT,11
         JNE    IOC2
         CMP    ES:[BX].P_FUN,60H
         JE     IOC_Q                   ; branch if IOCTL 11.96, QUERY MON
IOC2:    JMP    CMDBAD                  ; else indicate invalid request
IOC3:    JMP    ERRGEN
;
; Query monitor support (Category 11, function 96)
;
IOC_Q:   JMP    RETOK                   ; return OK to show monitor support
```

```
;
; Get printer status (Category 5, function 101)
;
IOC_S:   LES    DI,ES:[BX].P_DATA       ; verify access to data buffer
         MOV    AX,ES
         MOV    CX,1
         MOV    DX,0100H+VerifyAccess
         CALL   DEVHLP
         JC     IOC3                    ; branch if access denied
         XOR    BX,BX                   ; lock the data buffer segment
         MOV    DL,LockSeg
         CALL   DEVHLP
         PUSH   AX                      ; save lock handle
         PUSH   BX
         PUSH   DS                      ; convert to physical address
         MOV    AX,ES
         MOV    DS,AX
         MOV    SI,DI
         MOV    DL,VirtToPhys
         CALL   DEVHLP
         POP    DS
         MOV    DX,0100H+PhysToVirt     ; convert to virtual address
         CALL   DEVHLP
         JC     IOC4                    ; branch if error
         CLI                            ; begin critical section
         MOV    DX,IOLPT                ; get device status
         INC    DX
         IN     AL,DX
         AND    AL,0F8H                 ; save only important bits
         XOR    AL,48H                  ; toggle "acknowledge" and "error"
         MOV    AH,STATUS               ; include "timer", "flush", "wait"
         AND    AH,7
         OR     AL,AH
         MOV    ES:[DI],AL              ; return status in user's data area
         STI                            ; end critical section
         MOV    DL,UnPhysToVirt         ; release mapping
         CALL   DEVHLP
         POP    BX                      ; unlock data buffer segment
         POP    AX
         MOV    DL,UnlockSeg
         CALL   DEVHLP
         JMP    RETOK
IOC4:    POP    BX                      ; come here on error after lock
         POP    AX
         MOV    DL,UnlockSeg
         CALL   DEVHLP
IOC5:    JMP    ERRGEN
;
;
; Register a monitor (Category 10, function 64)
;
IOC_R:   CALL   GETPID                  ; get monitor process id
         JC     IOC3                    ; branch if error
         LES    SI,ES:[BX].P_DATA       ; get data area address
         MOV    DH,ES:[SI].R_POS        ; get postion code
         MOV    DI,ES:[SI].R_OUT        ; get output buffer offset
         LES    SI,ES:[SI].R_IN         ; get input buffer pointer
         MOV    AX,MONHAN               ; get monitor handle
         MOV    DL,Register             ; register this monitor
         CALL   DEVHLP
         JC     IOC5                    ; branch if error
         JMP    RETOK
;***********************************************************************
;
;
; S_RMV -- Remove (de-install) the driver
;
S_RMV:   MOV    AX,MONHAN               ; kill monitor chain
         OR     AX,AX
         JZ     RMV1
         MOV    DL,MonCreate
         CALL   DEVHLP
RMV1:    MOV    BX,IOINT                ; detach from interrupt
         OR     BX,BX
         JZ     RMV2
         MOV    DL,UnSetIRQ
         CALL   DEVHLP
RMV2:    JMP    RETOK
;
;***********************************************************************
;
```

```
; NOTIFY routine, called by monitor dispatcher when data is available.
;
NOTIFY  PROC    FAR
        TEST    MONR.M_FLAG,MF_OPEN
        JNZ     N_OPEN                  ; branch if "open" message
        TEST    MONR.M_FLAG,MF_CLOS
        JNZ     N_CLOS                  ; branch if "close" message
        TEST    MONR.M_FLAG,MF_FLSH
        JZ      N_WRT                   ; branch if "write" message
;
; Process "flush" message from the monitor chain
;
N_FLSH: AND     STATUS,NOT SF_FLSH      ; reset flush flag
        JMP     RETOK
;
; Process "open" message from monitor chain
;
N_OPEN: JMP     RETOK
;
; Process "close" message from the monitor chain
;
N_CLOS: JMP     RETOK
;
; Process "write" message from the monitor chain
;
N_WRT:  MOV     CX,MON                  ; get message length
        SUB     CX,MONHDR+2             ; subtract header and length word
        JBE     N03                     ; branch if no data
        XOR     SI,SI                   ; reset data buffer index
N00:    TEST    STATUS,SF_FLSH
        JNZ     N03                     ; abort if flush flag is set
        MOV     DX,IOLPT                ; get printer address
        INC     DX                      ; check status
        IN      AL,DX
        TEST    AL,80H
        JZ      N04                     ; branch if not ready
N01:    MOV     AL,MONR.M_DATA[SI]      ; send next byte
        DEC     DX
        OUT     DX,AL
        INC     DX                      ; strobe it out
        INC     DX
        MOV     AL,0DH
        OUT     DX,AL
        MOV     AX,10
N02:    DEC     AX
        JNZ     N02
        MOV     AL,0CH
        OUT     DX,AL
        INC     SI                      ; loop till all bytes sent
        LOOP    N00
N03:    RETF                            ; return to monitor dispatcher
;
; Come here to wait for printer interrupt
;
N04:    CLI                             ; begin critical section
        INC     DX                      ; enable printer interrupt
        MOV     AL,1CH
        OUT     DX,AL
        DEC     DX                      ; check status
        IN      AL,DX
        TEST    AL,80H
        JZ      N05                     ; branch if still busy
        MOV     AL,0CH                  ; disable printer interrupt
        INC     DX
        OUT     DX,AL
        STI                             ; leave critical section
        JMP     N00                     ; continue data output
N05:    PUSH    CX                      ; save registers
        PUSH    SI
        OR      STATUS,SF_WAIT          ; set "wait" status
        MOV     AX,WORD PTR IOTADD+2    ; use I/O table address as event id
        MOV     BX,WORD PTR IOTADD
        MOV     DI,WORD PTR IOTIME+2    ; load timeout value
        MOV     CX,WORD PTR IOTIME
        MOV     DX,Block                ; DH=0 for interruptable sleep
        CALL    DEVHLP
        CLI                             ; begin critical section
        JC      N07                     ; branch if timeout or unusual event
        TEST    STATUS,SF_WAIT
        JNZ     N05                     ; re-block if wrong event
N06:    MOV     DX,IOLPT                ; disable printer interrupt
```

```
        INC     DX
        INC     DX
        MOV     AL,0CH
        OUT     DX,AL
        STI                             ; end critical section
        JMP     N00                     ; go send more data
;
; Come here if wait times out or is aborted by unusual event
;
N07:    OR      STATUS,SF_TIME          ; set timer flag
        MOV     AX,WORD PTR IOTADD+2    ; use I/O table address as event id
        MOV     BX,WORD PTR IOTADD
        XOR     DI,DI                   ; set 500 ms timeout value
        MOV     CX,500
        MOV     DX,Block                ; DH=0 for interruptable sleep
        CALL    DEVHLP
        CLI                             ; begin critical section
        TEST    STATUS,SF_FLSH
        JNZ     N08                     ; branch if flush flag is set
        TEST    STATUS,SF_WAIT
        JNZ     N07                     ; repeat if still waiting
        XOR     STATUS,SF_TIME          ; reset timer flag
        JMP     N06                     ; continue data output
;
; Come here if operation is being flushed
;
N08:    AND     STATUS,NOT (SF_TIME OR SF_WAIT)  ; reset flags
        MOV     DX,IOLPT                ; disable printer interrupt
        INC     DX
        INC     DX
        MOV     AL,0CH
        OUT     DX,AL
        STI                             ; end critical section
        RETF

NOTIFY  ENDP
;*********************************************************************
;
; I_LPT routine, called by hardware interrupt dispatcher
;
I_LPT   PROC    FAR
        TEST    STATUS,SF_WAIT
        JZ      INT1                    ; ignore if not in "wait" mode
        XOR     STATUS,SF_WAIT          ; reset "wait" status
        MOV     AX,WORD PTR IOTADD+2    ; wake up the NOTIFY routine
        MOV     BX,WORD PTR IOTADD
        MOV     DL,Run
        CALL    DEVHLP
INT1:   CLI                             ; begin critical section
        MOV     AX,IOINT                ; signal end-of-interrupt
        MOV     DL,EOI
        CALL    DEVHLP
        CLC                             ; clear CF to claim interrupt
        RETF
I_LPT   ENDP
;*********************************************************************
;
; User interrupt routine, called by DOS application via interrupt 17H.
;
U_LPT   PROC    FAR
        STI                             ; enable interrupts
        PUSH    BP                      ; save all registers
        PUSH    ES
        PUSH    DS
        PUSH    DI
        PUSH    SI
        PUSH    DX
        PUSH    CX
        PUSH    BX
        PUSH    AX
        MOV     BX,CS:SAVEDS            ; set up DS for the driver
        MOV     DS,BX
        MOV     DX,IOLPT                ; get printer address
        OR      AH,AH
        JZ      U_WRT                   ; AH=0 to print AL
        DEC     AH
        JZ      U_INIT                  ; AH=1 to initialize
        DEC     AH
        JZ      U_STAT                  ; AH=2 to get status
U_RET:  POP     AX                      ; return from interrupt
        POP     BX
```

```
        POP     CX
        POP     DX
        POP     SI
        POP     DI
        POP     DS
        POP     ES
        POP     BP
        IRET
;
; Initialization routine saves DS here
;
SAVEDS  DW      0                       ; DS is saved here
;
; Come here to print the character in AL
;
U_WRT:  MOV     AL,1                    ; begin BIOS critical section
        MOV     DL,ROMCritSect
        CALL    DEVHLP
        MOV     CX,MONHDR+1             ; set monitor message length
        MOV     MONS.M_PID,0            ; use 0 for DOS process id
        MOV     MONS.M_FLAG,0           ; reset flags
        POP     AX                      ; put byte into buffer
        PUSH    AX
        MOV     MONS.M_DATA,AL
        LEA     SI,MONS                 ; send message to monitor chain
        MOV     AX,MONHAN
        MOV     DX,MonWrite
        CALL    DEVHLP
        XOR     AL,AL                   ; end BIOS critical section
        MOV     DL,ROMCritSect
        CALL    DEVHLP
;       JMP     U_STAT                  ; go get status
;
; Come here to initialize the printer
;
U_INIT:                                 ; same as status request
;
; Come here to get printer status
;
U_STAT: CLI                             ; begin critical section
        MOV     DX,IOLPT                ; get device status
        INC     DX
        IN      AL,DX
        AND     AL,0F8H                 ; save only important bits
        XOR     AL,48H                  ; toggle "acknowledge" and "error"
        MOV     AH,STATUS               ; include "timer", "flush", "wait"
        AND     AH,7
        OR      AL,AH
        STI                             ; end critical section
        POP     BX                      ; restore AH and set up AL return
        MOV     AH,BH
        PUSH    AX
        JMP     U_RET
U_LPT   ENDP
;
;****************************************************************************
;
; GETPID function, called to obtain current process id, which is returned
; in AX.  The function returns with CF set if an error occurs.
;
GETPID  PROC    NEAR
        PUSH    ES                      ; save regs
        PUSH    BX
        PUSH    DX
        MOV     AL,2                    ; address of process info pointer
        MOV     DL,GetDOSVar
        CALL    DEVHLP
        JC      PID1                    ; branch if error
        PUSH    AX                      ; get process info pointer in ES:BX
        POP     ES
        LES     BX,ES:[BX]
        MOV     AX,ES:[BX]              ; PID is first word
        CLC                             ; reset CF to indicate success
PID1:   POP     DX                      ; restore regs and return
        POP     BX
        POP     ES
        RET
GETPID  ENDP
;****************************************************************************
;
```

```
; Mark the end of the code segment.  Everything after this is discarded
; after initialization.
END_CS  EQU     $
;***************************************************************************
;
; S_INIT -- Initialize
;
S_INIT: MOV     AX,ES:[BX].P_HLPO           ; save DevHlp address
        MOV     WORD PTR DEVHLP,AX
        MOV     AX,ES:[BX].P_HLP1
        MOV     WORD PTR DEVHLP+2,AX
        LEA     AX,END_CS                   ; return CS size
        MOV     ES:[BX].P_HLPO,AX
        LEA     AX,END_DS                   ; return DS size
        MOV     ES:[BX].P_HLP1,AX
;
; Check for valid I/O configuration
        LES     DI,DWORD PTR IOTADD         ; get printer I/O address
        MOV     DX,ES:[DI]
        MOV     AL,IOINT1                   ; check if first parallel device
        CMP     DX,IOLPT1
        JE      INO1                        ; branch if yes
        MOV     AL,IOINT2                   ; check if second parallel device
        CMP     DX,IOLPT2
        JNE     INERR                       ; branch if no
;
; Initialize the I/O channel and device
;
INO1:   MOV     IOLPT,DX                    ; save I/O address
        MOV     BYTE PTR IOINT,AL           ; save interrupt level
        INC     DX                          ; set DX to control port
        INC     DX
        MOV     AL,8                        ; select and initialize
        OUT     DX,AL
        MOV     AX,1000                     ; delay at least 50 microseconds
INO2:   DEC     AX
        JNZ     INO2
        MOV     AL,12                       ; select, no initialize, no auto LF
        OUT     DX,AL
;
; Create monitor chain and save its handle
;
        PUSH    DS                          ; make ES:SI point to monitor buffer
        POP     ES
        LEA     SI,MON
        PUSH    DS                          ; save DS
        PUSH    CS                          ; make DS:DI point to notify routine
        POP     DS
        LEA     DI,NOTIFY
        XOR     AX,AX                       ; set AX=0 to create monitor chain
        MOV     DL,MonCreate                ; call DEVHLP using ES instead of DS
        CALL    ES:DEVHLP
        POP     DS                          ; restore DS
        JC      INERR                       ; branch if error
        MOV     MONHAN,AX                   ; save monitor handle
;
; Activate the hardware interrupt
        LEA     AX,I_LPT                    ; get interrupt routine address
        MOV     BX,IOINT                    ; get interrupt level number
        MOV     DX,SetIRQ                   ; DH=0 for non-shared interrupt
        CALL    DEVHLP
        JC      INO3                        ; branch if error
;
; Activate user interrupt 17H for printer BIOS
;
        LEA     AX,U_LPT                    ; get interrupt routine address
        MOV     BX,17H                      ; get interrupt number
        LEA     SI,SAVEDS                   ; get DS save location address
        MOV     DL,SetROMVector             ; set the interrupt vector
        CALL    DEVHLP
        JMP     RETOK
;
; Come here on initialization error
INO3:   MOV     AX,MONHAN                   ; kill monitor chain
        MOV     DL,MonCreate
        CALL    DEVHLP
INERR:  JMP     ERRGEN
;
S_LPT   ENDP                                ; end of main proc
_TEXT   ENDS                                ; end of code segment
        END                                 ; end of module
```

It's good for your system!

Vitamin C

"If you need source code, make sure your wallet is wide open or get VITAMIN C.

Picking the best value package is hard... If you're a source code fanatic like me, VITAMIN C is preferable."

- Computer Language, June, 1987

Fast, flexible, versatile, reliable. Vitamin C delivers the vital combination software professionals demand to produce superior applications in dramatically less time. Highly efficient, professionally crafted C code provides lightning fast displays required by today's window intensive programs.

High level functions provide maximum productivity and require little supporting code. Extended versions of these routines add flexible control over specific details when necessary. Plus, Vitamin C's versatile, open ended design is full of hooks so you can intercept and plug-in special handlers to customize or add features to most routines.

VCScreen, our screen painter / code generator speeds your development even more! Simply draw your input forms using our interactive design editor and generate perfect C source code ready to compile and link with the Vitamin C library.

- ☑ Professional C function library
- ☑ 30 day money back guarantee
- ☑ Multiple bullet proof windows
- ☑ Easy full screen data entry
- ☑ Unlimited data validation
- ☑ Context sensitive help manager
- ☑ Menus like Lotus and Mac
- ☑ Programmable keyboard handler
- ☑ Text editor routines
- ☑ No royalties or runtime fees
- ☑ Library source included FREE
- ☑ Free technical support
- ☑ Free BBS at (214)418-0059
- ☑ Supports all major compilers including Microsoft 5.0
- ☑ VCScreen code generator too!
- ☑ UNIX version avaialable, call for details

Vitamin C $225
Includes all source code FREE! For IBM PC, XT, AT, PS/2 and true compatibles. Specify compiler when ordering.

VCScreen $99.95
For IBM PC, XT, AT, PS/2 and true compatibles. Requires Vitamin C.

We ship UPS. Please include $3 for ground, $6 for 2-day air, $20 for overnight, or $30 if outside the U.S. Texas residents add 7 1/4 % sales tax. All funds MUST be in U.S. dollars drawn on a U.S. bank. Visa & MasterCard accepted.

ORDER NOW!

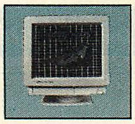
(214)416-6447

creative PROGRAMMING
Box 112097 • Carrollton, Texas 75011

Windows • Data Entry • Menus • Help • Text Editing
Plus... All Source Code FREE!

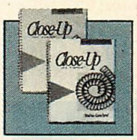
PRODUCT WATCH

Reviews and Updates



MULTISYNC PLUS
NEC Home Electronics (USA), Inc.

MULTISYNC II
NEC Home Electronics (USA), Inc.



CLOSE-UP
Norton-Lambert Corporation

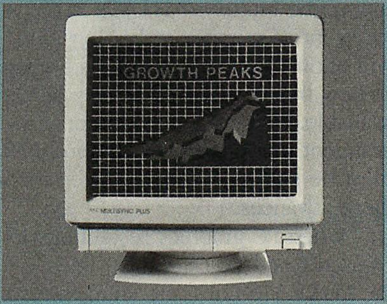
BEYOND·BAT
VM Personal Computing, Inc.

UPDATE

MULTISYNC PLUS

NEC Home Electronics (USA), Inc.
Personal Computer Products Division
1255 Michael Drive
Wood Dale, IL 60191-1094
800/632-7638; 312/860-9500

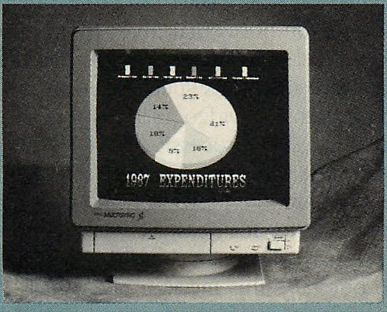
PRICE: $1,395



CIRCLE 329 ON READER SERVICE CARD

MULTISYNC II

PRICE: $899



CIRCLE 330 ON READER SERVICE CARD

With the advent of PS/2, IBM upped the ante in the personal computer graphics arena by adopting a minimum display resolution of 640-by-480 pixels (the enhanced resolution of the previous generation), and abandoning transistor-transistor logic (TTL) monitors in favor of analog models. The time therefore is ripe for the next step in PC graphics resolution: either enhancing the PS/2 standard Video Graphics Array (VGA) to 800-by-600 pixels or providing a less expensive alternative to the 8514A resolution of 1,024-by-768 pixels. Achieving this level of graphics requires both a video adapter and a monitor.

NEC has responded to the call for a high-resolution monitor with the JC-1501, the MultiSync PLUS Color Monitor. Whereas the original MultiSync Color Monitor, used in conjunction with a third-party enhanced graphics adapter card (such as the VEGA Deluxe by Video-7 Incorporated), could expand the resolution of the PC beyond the initial limitations of the IBM EGA to 640-by-480 pixels, the new MultiSync PLUS can provide a resolution of 960-by-720 pixels. In addition, NEC has upgraded and repackaged the original MultiSync (JC-1401) as the MultiSync II (JC-1402). The original MultiSync monitor was reviewed in an earlier issue ("Synchronizing Graphics Standards," John C. Blair Jr., May, 1987, p. 146.)

Both the MultiSync PLUS and the MultiSync II are housed in similar external packaging, which is considerably improved in terms of human factors compared with the original MultiSync. On both updated machines, the bezel is shallower, and the screen is flatter. The controls that the user adjusts frequently are located within easy reach at the bottom of the display. The power switch is now in the front of the unit. The video connections and video signal controls, which are presumably less frequently manipulated, are on the rear of the device. The heights of both monitors are at eye level when they are freestanding on a desktop. The footprints are the same.

The MultiSync II is the same monitor as the MultiSync and has the same functionality, except that it is VGA-compatible. The vertical sync frequency ranges from 50 to 80 Hz, whereas the MultiSync's specifications called for a narrower range, 56 to 62 Hz. The dot pitch, misconvergence, display bandwidth, and active display area are identical to the MultiSync (see table 1). NEC includes a 9 pin to 15 pin conversion adapter to allow cabling of the monitor to the PS/2 VGA. NEC has done well with the improvements engineered into the MultiSync II.

The MultiSync PLUS boasts a higher bandwidth and horizontal scan rate as well as greater horizontal and vertical resolution than both the original MultiSync and the MultiSync II (see table 1). With these improvements as well as a larger picture size and more conveniently positioned controls, NEC has positioned the MultiSync PLUS as the display for the next generation of high-resolution PC graphics adapters.

The increased video bandwidth allows more dots to be displayed per second without blurring, while the increased horizontal sync frequency allows more horizontal lines to be displayed per second. In addition, the MultiSync PLUS can display a flicker-free image, because its high vertical sync frequency (50 to 80 Hz) allows an increased refresh rate. The horizontal and vertical sync frequency limits are translated into horizontal and vertical pixel resolutions of 960 by 720.

Unfortunately, the increase in display bandwidth comes with a price: the NEC MultiSync PLUS is not compatible with the lower-resolution display controllers, such as the Color Graphics Adapter (CGA) and the Monochrome Display Adapter. The absence of CGA graphics is noticeable, causing a distorted screen when Microsoft Windows displays its logo screen. However, this deficiency is not a significant loss because most graphical processes have migrated to the higher vertical resolution afforded by the Enhanced Graphic Adapter (EGA) standard.

All medium- and high-resolution display adapters, both analog and digital, can drive the MultiSync PLUS. In

TABLE 1: *Monitor Specifications*

	MULTI-SYNC PLUS	MULTI-SYNC II
Bandwidth (MHz)	55	30
Picture size (inches)	14	13
Horizontal sync frequency (kHz)	21.8–45	15.5–35
Vertical sync frequency (Hz)	50–80	50–80[a]
Maximum horizontal pixel resolution	960	800
Maximum vertical pixel resolution	720	560
Shadowmask pitch (mm)	0.31	0.31

[a] *The vertical sync frequency is higher than in the original MultiSync. All other MultiSync II values are the same as the original.*

The MultiSync PLUS has a higher bandwidth, horizontal and vertical sync frequencies, and resolution than its predecessor. The MultiSync II has the same specifications as the original MultiSync, except for a higher vertical sync frequency.

addition to the standard D shell connector for TTL and some analog display adapters, the MultiSync PLUS also has a 9-pin D-sub to 15-pin D-sub connector for use with IBM PS/2 Computers. A NEC custom cable is provided for PS/2 compatibility and an optional 9-pin D-sub to 4 BNC cable is available for BNC compatibility.

Except for the increase in the CRT size, the graphical characteristics of this display are the same as its predecessor. The aspect ratio of the MultiSync PLUS monitor is 4:3, a ratio now found commonly on PC displays. The dot pitch is 0.31 mm, and the matrix is the black-dot-type. The phosphors are nonlong persistence. All these features make the monitor easy on the eyes. While the misconvergence of the blue components of white pixels noted on the MultiSync has been lessened, it has not been completely eliminated in the MultiSync PLUS. When the video signals are removed, the monitor goes blank rather than generating a white screen, as is the case with monitors for the MultiSync and MultiSync II.

One slight problem, which was reproduced on another MultiSync PLUS, was noted with the display: the upper two lines of the screen are spaced more tightly than the remainder of the display, resulting in a distortion in their appearance. This distortion could not be eliminated either by using vertical position controls to position the image on the screen or by changing the display resolution.

Another drawback is related to the fan that NEC has introduced to cool the MultiSync PLUS. Cooling is necessary because of the increase in heat-producing electronics in the new machine and also to avoid failures similar to those reported with the original MultiSync. However, the fan is quite noisy.

Despite the absence of video boards that can take advantage of its increased vertical and horizontal resolution, the MultiSync PLUS is at the forefront of monitor technology and will allow its owners to take advantage of new board advances as they become available. However, the additional investment that this monitor represents must be balanced with the need for these advances.

When PC graphics resolutions increased from the CGA's 200 lines to the EGA's 350 lines, the jump in resolution was a stunning improvement in visual appearance. Moreover, the graphics performance of the host CPU was not a factor. Similar conclusions can be drawn for the boost from the EGA's 350 to 480 horizontal lines. However, the increase from 480 lines to 800 lines or more mandates special, and probably expensive, video adapters that offload graphics processing from the host. Many users may be satisfied with 480-line resolution. Moreover, the length of time it takes for software to catch up with hardware may cloud the decision to purchase this monitor and a high resolution graphics adapter.

Overall, the NEC MultiSync PLUS represents an improvement in monitor standards. But compared with other monitors in its price and performance class, such as Sony's MultiScan, its performance is at best average. Its $1,395 price may not justify the improvement—at least until video adapters become more readily available to exploit its features. At present, the MultiSync II, at $899, may be the monitor of choice for most PC applications.

—JOHN COCKERHAM

FROM THE MAKERS OF CARBON COPY™

SpecTerm™ is a terminal emulation software package which transforms your PC into a complete workstation by giving you access to larger, more powerful computers and information services. Running on an IBM PC, XT, AT, PS/2, or compatible, **SpecTerm** makes it simple to use your local keyboard and screen exactly as you would use a terminal attached directly to the host computer. Now you can access a large database, run a VAX application, get the latest stock quotes from Dow Jones, make airline reservations, exchange information with your peers, and download any files you need to your own PC with the added benefit of your own computing power.

And that's not all! **SpecTerm** has enhanced graphics capabilities. Providing emulation properties for both Tektronix and ReGIS Graphics on your PC. Now, make your PC a much more powerful communications medium, one that is far more flexible and economical than any single terminal it emulates.

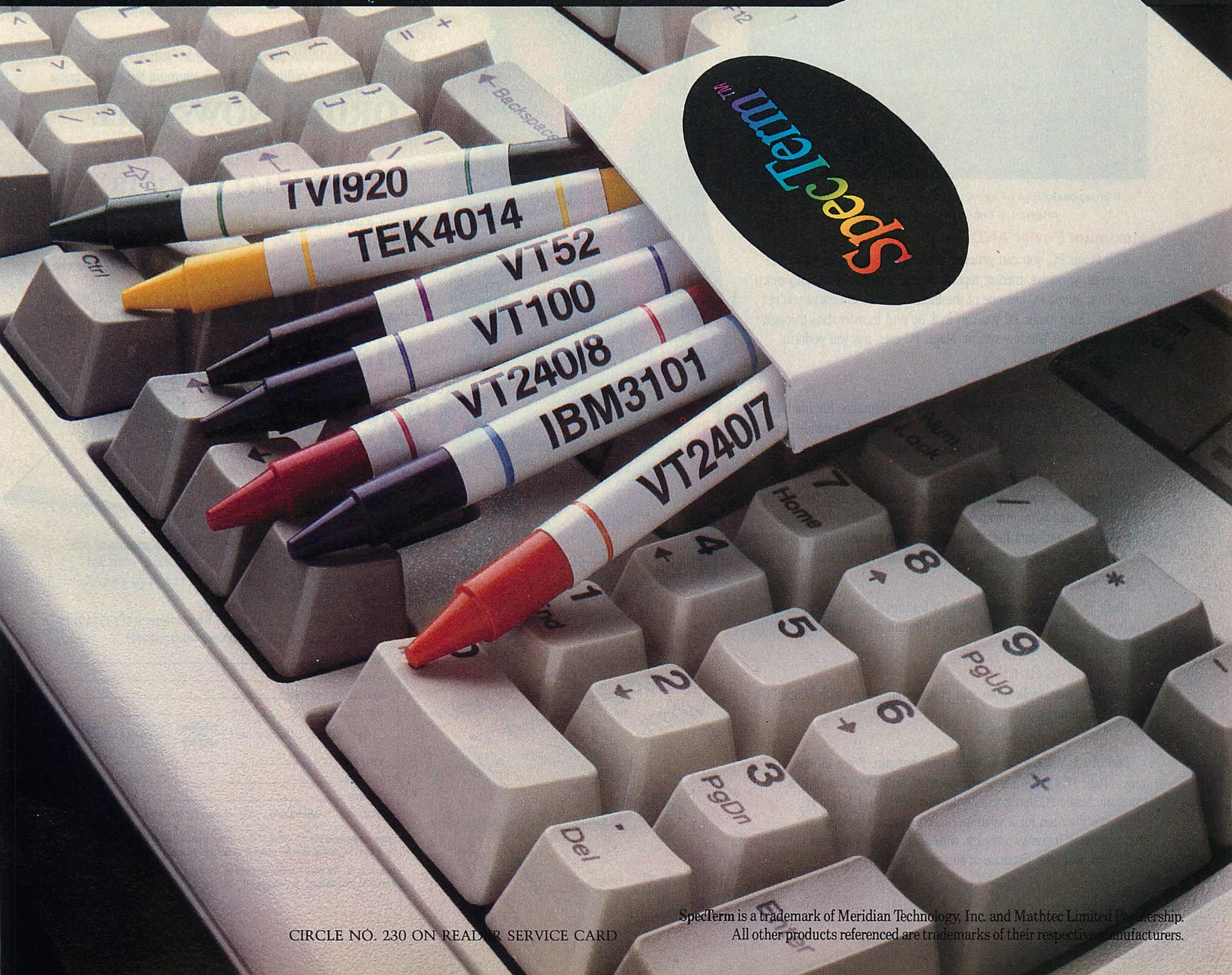
SpecTerm, from the makers of Carbon Copy PLUS, includes many of the foremost features found in our other communication software products. These include auto script file execution, support of high-speed modems, full COMM Support (1, 2, 3, and 4), Kermit and X-Modem protocols, and a powerful Command language.

SpecTerm™

Terminal Emulation for Your PC!

VT-240/7 • VT-240/8 • VT-100 • VT-52 • TEK 4014 • TVI-920 • IBM3101

MERIDIAN TECHNOLOGY INC
7 CORPORATE PARK ▪ SUITE 100 ▪ IRVINE, CALIFORNIA ▪ 92714
(714) 261-1199



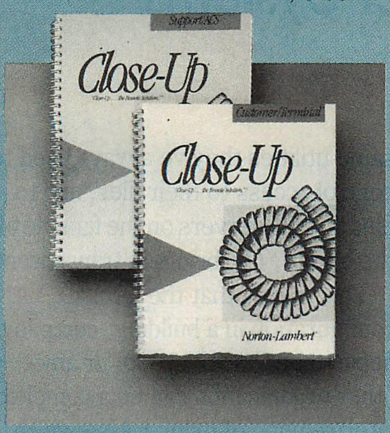
CIRCLE NO. 230 ON READER SERVICE CARD

SpecTerm is a trademark of Meridian Technology, Inc. and Mathtec Limited Partnership. All other products referenced are trademarks of their respective manufacturers.

CLOSE-UP 3.0

Norton-Lambert Corporation
P.O. Box 4085
Santa Barbara, CA 93140
805/964-6767

PRICE: Support/ACS, $245;
Customer/Terminal, $195



CIRCLE 332 ON READER SERVICE CARD

Is it live or is it Close-Up? This is a question that is provoked by Close-Up, the communications software from Norton-Lambert Corporation that supports realtime "videotaping" of remote sessions with a unique record and playback capability. When this record-to-disk feature is combined with Close-Up's script language, innovative and effective communications applications are possible.

The primary role of Close-Up is to access a remote PC. As a member of the class of remote-access products that includes Carbon Copy and PCANYWHERE, Close-Up is not the first software of its kind, but it has capabilities not found in similar products. Close-Up provides a feature-rich remote communications environment with strong development and customization functions and excellent documentation. There is so much in the Close-Up package that only the most prominent features were tested and are discussed here. Along with the recording and task language functions, Close-Up provides terminal emulation, background file transfer, a transaction log for time and billing, a movable chat window, extensive command-line options, and a high level of communications performance.

A few of the many possible remote applications for Close-Up are: automated code patching, multisite version updating, keyboard conferencing with realtime replay record, automated file transfer, and remote network support, as well as other automated remote functions that involve passing keystrokes to the remote PC application under program control.

Close-Up is largely a menu-driven program; the pop-up selections work smoothly and they are easy to use. Norton-Lambert has divided the package into two programs: SUPPORT/ACS and CUSTOMER/Terminal. The site that wishes to take over a remote PC runs the SUPPORT program, while the site receiving the call runs the CUSTOMER program. The CUSTOMER program is a terminate-and-stay resident (TSR) program that waits in the background for a call from SUPPORT (this is similar to Carbon Copy's CC and CCHelp programs). CUSTOMER requires 49KB to 66KB of memory, depending on how it is configured.

Either the CUSTOMER or the SUPPORT parts of the product can originate a call by selecting first the Phone pull-down menu and then the Dial submenu. If names and phone numbers have been entered in the dialing directory, the user may enter a name for the site instead of the phone number. After the call is made, the two computers are connected; if passwords are defined, the user is asked for the password.

Once connection is made, the screen updates of the computer running CUSTOMER are directed through the phone line to the PC running SUPPORT. In addition, the output from the SUPPORT computer keyboard is directed to the CUSTOMER screen. The effect of these redirections is that the application running on the CUSTOMER PC is displayed on both local and remote screens, and anything entered on either keyboard will be displayed. The SUPPORT user has full use of the CUSTOMER computer just as a local user would, except that it is somewhat slower, depending on the speed of the modems. CGA graphics and full use of the IBM keyboard is supported. If a private session is desired, the SUPPORT side can disable the screen and keyboards of the CUSTOMER side.

The simplicity of making a call with Close-Up cannot be overstated. Most communications software requires much time spent on framing parameters such as data bits, stop bits, and parity, as well as ID numbers and passwords. With Close-Up, the name of the remote location is selected, the computers connect, and the session begins. As with any communications software, the possibility of connection problems exists, but Norton-Lambert has made a great effort to reduce these with Close-Up. The speed of the support-side screen update is comparable to other remote products and is protected from line noise and other interference with a cyclic redundancy check (CRC) error-checking algorithm.

Close-Up makes substantial use of hot keys during the communication session. The user can modify the hot-key combinations, which are used to access main menus, the terminal emulator menu, and the chat function. In chat mode, the two users can chat with their keyboards in a window that is overlaid on top of the application that is running. Unlike the chat windows in competing remote-access software, Close-Up's chat windows can be moved with the arrow keys so as not to obscure areas of the screen relevant to the chat session.

The menu for the SUPPORT side of Close-Up has a selection that allows the user to exit to DOS and run an application while still connected to the remote PC. When this function is used, it is important that the local application does not affect the COM port handling the Close-Up connection, or the line will be dropped. If the SUPPORT user has exited to DOS to run an application or interact with the DOS command processor, the remote user or program can send a Ctrl-G, which rings the bell as an audible alert signal. This alert works for most other remote packages as well.

While running either SUPPORT or CUSTOMER, Close-Up file transfer is accomplished through the menu. Once invoked, file-transfer security rights are checked, and the user is presented with a submenu for sending or fetching (receiving) files. The menu method of initiating a transfer is a mixed blessing, which involves the classic trade-off of menu versus command line user interface. Whereas specifying transfer and entering the filename is quite easy with the menus, it is not possible to specify full DOS pathnames on the remote system. Instead, the remote default directory must be changed (through the menu) before a file can be transferred to, or fetched from, the directory.

File transfer generally proceeds very smoothly, and the transfer status display is impressive. Close-Up compresses files during transfer for very fast transmission. This is done by a proprietary, block-oriented error-detection technique described in the manual as "fast sliding window (CRC-16) block numbered FSV error-checking and correcting protocol." When file transfer is

Open Protocol Technology

The Better Bridge Between Computer Systems

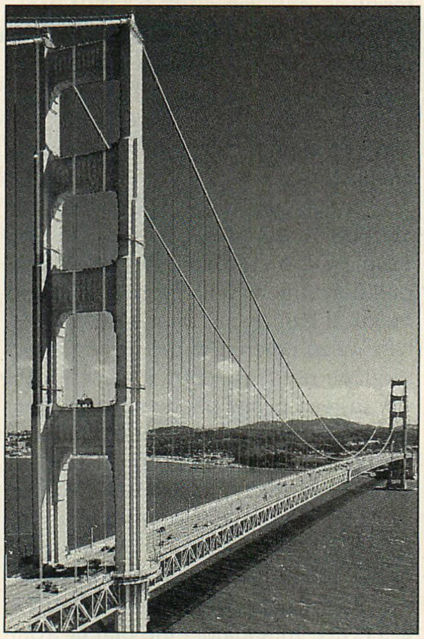
A bridge is a device for carrying things over a river. So is a ferry. But when you're driving across a river there is a great difference between taking a bridge and taking the ferry. When you drive across by bridge, you just continue doing what you were doing—the rules of the road don't change. When you cross by ferry the rules change—you park until the ride ends.

A bridge, in the world of NetWare,® connects across the great gulf between two computer networks. Like a river bridge, a NetWare bridge is an ideal way to travel. It connects two networks without changing the rules of the road—an application program working with a bridge, for instance, doesn't need to know which network the data is coming from.

When files travel across a NetWare bridge, the ground rules remain the same regardless of which physical network the files come from. This convenient lack of change is called "transparency" or "seamlessness"—the user or application "looks" across the bridge and sees no difference in the rules for dealing with files.

With NetWare, as many as fifteen bridges can be looked across transparently. In practice, this means that NetWare-based LANs offer great flexibility in building multiple LAN environments. A company's departmental LAN environments can grow independently without sacrificing expense or performance when the time comes to join them together.

For instance, a company could start with a department LAN consisting of Ethernet® hardware in the



"The concept of the transparent bridge is the heart of Novell's Open Protocol Technology."

office. Shortly thereafter, the production people could decide a Token-Ring LAN suits their needs, and install one on the factory floor. When the time comes to link them together, it would merely be a matter of adding an adapter board, stringing some cable, and quickly reconfiguring NetWare to activate the bridging function.

Once the bridge is forged, users on both sides continue to use the same utilities they've always used to control access to their files, and to files and file servers on the far side of the bridge. The bridge is transparent.

This means that the ultimate LAN configuration in a building, company, department, work group, or any other organization can be a complex interweaving of many LANs, which are physically different but uniform in their software and their response to user commands.

Extending the Bridge Concept

The concept of the transparent bridge goes beyond just connecting LANs to LANs. It is the heart of Novell's Open Protocol Technology (OPT). The goal of OPT is to have other computer systems connect to a LAN and remain just as transparent to applications and users as NetWare bridge connections are now.

Novell is in the bridge-building business. It is our intent to connect wider and wider circles of networks. Novell's purpose for OPT is to ease the journey for application programs and end users by allowing the rules of the computer road to remain the same; regardless of where users travel through the world of networks.

Call 1-800-LANKIND if you would like to receive a copy of Novell's full Universal NetWare Architecture report.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVLROPT3.

NOVELL.

active, the display shows the current compression ratio, along with the overall ratio, number of bytes remaining to transfer, and number of bytes transferred. A realtime graph of percent completion is also shown.

While connected with the remote PC, the user can record screen images through the record function. A single screen or an entire session can be recorded as if on videotape. The screens are saved in a file that can be displayed later, either as a slide show or a movie. During the viewing process, the arrow keys control the speed of the playback, and the space bar is used to pause the display. The record function is an impressive feature that, when used in combination with task scripts, has a wide range of uses in areas of software maintenance, network support, and mission-critical remote operations that require a record of each keystroke.

The CUSTOMER portion of Close-Up includes TERMINAL, a very useful terminal-emulation feature. TERMINAL is a TSR communications program requiring an additional 13KB of memory and suitable for communication with on-line services and other remote hosts as well as PC-based bulletin boards. The user can exit to a foreground application while the TERMINAL session is conducting file transfer or other on-line operations in the background. Cyclic reduncancy check (CRC) and checksum XMODEM protocols are supported. In our tests, Close-Up TERMINAL executed a large XMODEM file transfer in the background, with WordPerfect in the foreground; little or no degradation for either process was apparent. When the transfer was completed, a small pop-up window was superimposed on the WordPerfect screen to indicate a successful file transfer. TERMINAL has an automatic parity and data-bits feature that greatly simplifies the determination of transmission parameters for a diverse systems.

The script language of Close-Up, called Automated Communications System (ACS), is particularly useful for unattended or batched remote operations. ACS allows users to write small programs (scripts), called task files, that are executed during a Close-Up session. The task files can direct an unattended PC to wait until a specified time, dial up a remote system, run programs, enter keystrokes (including function keys), check areas of the screen against a particular value, and take a snapshot of the screen if the field does not match. The realtime rec-

ord feature can be made active to create a disk file "videotape" of the unattended session.

One use for Close-Up's versatile ACS script facility is the remote update of files too large to conveniently move between systems. Rather than receiving files, performing the update locally, and returning files to the remote system, ACS can direct the local PC to analyze and modify data with the remote systems screen editor or debugger. Before disconnecting, ACS can run applications remotely to test the integrity of the updated files. One function of the ACS will compare values at specific row and column positions on the screen to values in the script. If values on the screen do not compare correctly, an exception routine can be initiated. This comparing ability gives ACS the intelligence to know if anomalies occur in an automated remote session. The command set of ACS is small and does not comprise a full communications language, but some unique applications are possible for it.

Close-Up's SUPPORT program includes a syntax-checking function to examine a task file; it generates error messages based on the syntax of the task file. If errors are found, the user can edit the file, recheck it, and then run the task.

The installation of Close-Up is very simple: it consists of copying a file onto the working diskette drive, hard disk, or file server. Norton-Lambert includes just one executable on the SUPPORT and CUSTOMER disks. Setup and configuration of COM ports, colors, and other options are performed from the main menu and can be modified to meet different requirements.

The manuals for Close-Up are particularly well done. They are neither over- or under-documented, and they are clearly laid out with sections on modems, cables, configuration recommendations, software compatibility findings, error messages, and troubleshooting techniques.

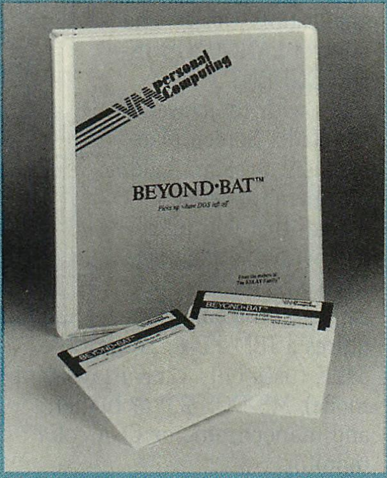
Although not initially as well known as other remote products, Close-Up is rapidly gaining acceptance. Novell, Inc., Ashton-Tate, and 3 COM Corp. have entered into licensing agreements and use Close-Up widely for corporate communications. In spite of the frustrations the product's menu systems may cause, the useful array of communications functions and unique features of this package make it well worth the price.

—CHARLES ROSE

BEYOND·BAT

VM Personal Computing, Inc.
41 Kenosia Avenue
Danbury, CT 06810
800/847-3529; 203/798-3800

PRICE: $99



CIRCLE 333 ON READER SERVICE CARD

BEYOND·BAT, from VM Personal Computing, Inc., enhances the capabilities of DOS, and it also offers transparency from DOS. In addition to a full screen editor and a library facility, BEYOND·BAT provides a powerful script processor that enables a user to create or enhance system utilities; in addition, it offers a screen management facility that allows a user to develop customized screens for various applications. The screens can then be used by script files to provide a user-friendly operator interface. All script and screen panel files associated with a specific application can be placed in a library for future use.

BEYOND·BAT will run on any IBM or fully compatible PC running DOS 2.0 or greater. It is installed by copying the files on the BEYOND·BAT system diskette to a specified directory on the hard disk or to a work diskette, and entering the drive and path to the PATH statement in the AUTOEXEC.BAT file. The memory requirement of BEYOND·BAT ranges from 65KB to 172KB, depending on the configuration, which is either nonresident, DOS extension, or totally resident. In nonresident mode, BEYOND·BAT is invoked from the DOS prompt; the script to be executed is supplied as an argument, such as:

```
C> bb sample.bat
```

In DOS extension mode, a portion of BEYOND·BAT is loaded as a terminate-and-stay-resident (TSR) program. Therefore, the invocation of BEYOND·BAT does not need to take place at the DOS prompt; for example,

```
C> sample.bat
```

These configurations are recommended for hard-disk systems. The totally resident configuration loads the entire BEYOND·BAT program into RAM. The invocation is similar to the DOS extension mode, but takes up considerably more RAM (172KB). This configuration is useful for diskette drive systems.

Script files, like DOS batch files, contain all commands and parameters that a user would need to enter from the keyboard to accomplish a specific task. They offer the following additional features: return codes on built-in functions; subroutine calling, variables, and string handling; and arithmetic, extended conditional, and logic functions. Script-file sizes are limited only by the available PC memory.

The script files are composed of the following four element types: variables, functions, commands, and labels. Variables can have names up to 9 characters in length and values up to 255 characters in length. The variables allow the user to save information in named memory cells. System variables have reserved variable names such as &DATE (which contains the current date); user variables can be defined by the user. Built-in functions allow the user to perform data manipulations easily on specified sets of data. BEYOND·BAT provides over 40 functions that perform a wide range of services—from data conversion to string manipulation to DOS interfacing.

The script commands that are supported by BEYOND·BAT, like those in most programming languages, allow complete program control flow, logic generation, and screen and file input/output (I/O). Commands and subcommands can be a maximum of 255 characters long; subcommands and condition statements can have a maximum of 128 characters. Labels are place markers that can be interspersed throughout the script source for script execution control or documentation purposes.

Script files have several other noteworthy features. For example, the script command, strace, displays the script commands as they are executed and is therefore a handy debugging tool. Script files have the ability to pass parameters to applications invoked from within BEYOND·BAT., and to examine return codes from these applications. Finally, the user can terminate

any BEYOND·BAT script file by pressing the Ctrl-Break key sequence.

Script files can have any file extension except .COM or .EXE. They *must* have an extension if they are to be executed by the script processor. If the extension is not .BAT, the script file must be invoked by using the **bb** (nonresident mode) command. These files are uniquely identified to the script processor as a BEYOND·BAT script file by having an asterisk in column one of the first line. The script files are executed in the same manner as are the DOS batch files.

The feature that makes BEYOND·BAT truly unique is that it allows a user to design customized screens (or panels) easily. Panels can present information to, and retrieve information from, the screen in an organized, consistent, and professional manner. In addition, the panels can be used to provide inherent error checking (for example, range checking on input) as well as complete DOS transparency.

A screen panel is an ASCII file that defines the position, contents, and display attributes of the screen text, input, and output fields. A panel can be created using any editor that stores its results as ASCII text. Each screen panel consists of a comment, attribute, body, and initialization section. The comment section contains all comments about the panel. The attribute section defines the characters used to indicate the various display attributes used in the body section (for example, **$ text high** means that all text that follows the dollar sign will be displayed in high intensity). The body section defines the 24-line-by-80-column screen image (or layout). Graphics characters can be used in the body section. The initialization section defines which user variables will be associated with the dynamic input and text fields (fields that change during script execution). VM Personal Computing provides many examples of panel definitions in the BEYOND·BAT manual; sample applications supplied on diskette aid in understanding and developing screen panels.

The BEYOND·BAT editor can be used to develop scripts or as a general editing facility. The editor is tailored for script development in that it allows the user to toggle between the editing and execution of a particular script.

In addition, BEYOND·BAT provides a functional library facility that can be used to logically group scripts, panels, or data into a single entity for the purposes of organization.

BEYOND·BAT also comes with a sample applications diskette that has some interesting and useful utilities. WAIT.COM is a program that can be used to wait until a specified time to execute a specific program; unfortunately, it is not a TSR program, so the user's computer is not free to perform alternate tasks. FFIND.BAT displays the directory in which a specified file is located. TREEDSP.BAT displays the tree structure of the specified drive. MENUDOS.BAT provides DOS capabilities through a menu-driven interface (rather than from the DOS command prompt line), thus providing DOS transparency.

BEYOND·BAT is not without some minor bugs. One problem occurred when it was run in total resident mode on an IBM XT compatible. If BEYOND·BAT is invoked from the command line (via **bb**), the computer locks up, necessitating a hardware reboot. The solution is that when running in total resident mode, the user should never need to invoke BEYOND·BAT from the command line. (This problem did not occur on an IBM XT or AT.) A warning message about this potential problem would be useful. The second problem is with the print screen script command, which is supposed to work exactly like the PrtSc key, but does not.

Technical support is available to users for a period of 30 days, starting from the first technical support call. Unfortunately, the customer support number is not toll-free. Even though this package is available at an attractive price, the user should not be penalized for having a problem with the package.

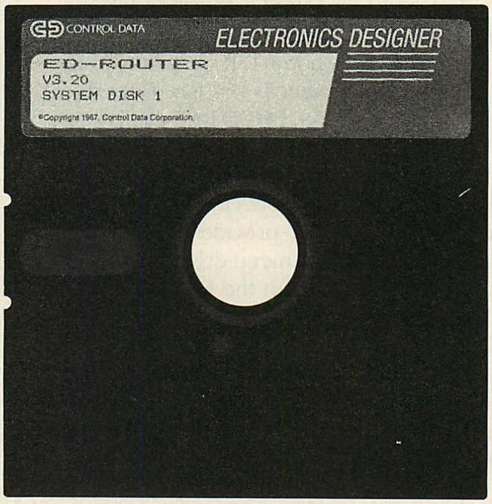
The documentation supplied with the product is organized, clear, and concise. It includes both a table of contents and an index. The descriptions are detailed and the examples are thoroughly explained. A quick reference card that lists the functions/commands and a list and description of runtime errors would make a handy addition.

A final note of interest—the BEYOND·BAT program was run in all three modes with SideKick and other TSR programs and experienced no conflicts with these programs.

BEYOND·BAT provides the tools to generate a wide range of applications. Whether you are responsible for system configuration in a shared PC environment or just want to develop automated system tools with an elegant front end, BEYOND·BAT would be a valuable addition to any software library.

—JOSEPH KRALOWETZ

This Software Won't Run.



Without This.

And that's the way Control Data wants it. And a host of other big and not-so-big software developers who use our Software Sentinel. To make sure their successful software is protected. So it stays successful.

We've become used to being seen with top-notch software. A lot of the reason is how our engineers designed the Software Sentinel family. For the developer and the user. To make it the most technologically foolproof yet easiest to use software protection key you or Control Data can get.

For users of ED-Router and the thousands of other programs we're protecting, it's a cinch to get up and running. They simply plug the Software Sentinel into the PC's parallel printer port. That's it. How much easier can it get? Users can even make unlimited backup copies. And run them wherever and whenever they need to—as long as they have the Software Sentinel key.

As long as we're talking success, there's something else you should know. Right now, our engineers are putting the finishing touches on a new microprocessor that we'll use in a whole new generation of software and data protection products. With the same high-performance and high-reliability of our past successes but with many new features. Which no doubt is going to make life even easier for you and Control Data.

The Software Sentinel. Making sure software developers stay successful with the successful software they've developed.

For more information on the Software Sentinel family, contact Rainbow Technologies, 18011-A Mitchell South, Irvine, California 92714; or call (714) 261-0228.

Software Sentinel Features:

- *Runs under DOS and Xenix*
- *Uses algorithm technique, never a fixed response*
- *Minimal implementation*
- *Higher level language interfaces included*
- *Transparent operation*

RAINBOW TECHNOLOGIES

18011-A Mitchell South, Irvine, CA 92714 • (714) 261-0228 • TELEX: 386078 • FAX: (714) 261-0260

CIRCLE NO. 157 ON READER SERVICE CARD

TECH NOTEBOOK

A forum for sharing solutions to technical problems

1 SETTING THE VGA

2 COLOR TEXT

The several subjects presented here each month will not always be related, but will cover the most interesting submissions, discoveries, and technical solutions that are available at any given time. This month, however, the two items do have a common theme—they both deal with making better use of video adapters in text and graphics modes. The first item, from Richard Wilton of Los Angeles, solves a potential incompatibility between VGA and EGA display adapters in text mode. The second, from Ben Myers of Harvard, Massachusetts, presents a method for specifying both background and foreground colors for text written in graphics modes. It is the first program published in *PC Tech Journal* that is written in Turbo Pascal 4.0.

1 SETTING VGA RESOLUTION

The Video Graphics Array (VGA) subsystem in IBM's PS/2 Models 50, 60, and 80 is highly upward compatible with its predecessor, the Enhanced Graphics Adapter (EGA). At a hardware level, the VGA's control register set incorporates EGA-compatible data formats and functions. Furthermore, the ROM video BIOS provides a superset of the functions that are available via interrupt 10H on the EGA.

Nevertheless, you can still encounter some incompatibilities on the VGA when you run a program written specifically for the EGA. One of the more common problems, and one of the simplest to correct, is related to a difference in the default text modes on the two video subsystems. The VGARES.C program (listing 1) helps you work around this difference.

On the EGA, the default text mode—that is, the video mode in which the EGA is configured by the video BIOS when the system is powered up—depends on the setting of four configuration switches. IBM's EGA and most compatibles have a set of four hardware DIP switches, while some EGA clones use software switches in nonvolatile RAM. You set the switches when you configure the EGA to use a particular video monitor. With color monitors, the switches may be set to select a default text mode whose vertical resolution is either 200 or 350 scan lines. Older monitors designed for use with the IBM Color Graphics Adapter (CGA) can use only 200-line modes, while EGA-compatible monitors can handle either the 200-line or the 350-line modes.

In contrast, the vertical resolution of the VGA's default text mode is 400 scan lines. To maintain EGA compatibility, however, the VGA also supports 200-line and 350-line text modes. The video BIOS configures the VGA for any of these three vertical resolutions, depending on the current settings of two flag bits in a reserved BIOS data byte at address 40:89H.

In most cases, the increased vertical resolution of the VGA does not affect the performance of well-behaved EGA-compatible applications, that is, those that allow the ROM BIOS to handle all screen I/O. But programs written specifically to the 350-line resolution of the EGA may not run properly in the VGA's default text mode. For example, a program that displays characters 8 pixels high might assume that there are 43 (350 ÷ 8) character rows on the screen, whereas on the VGA a total of 50 (400 ÷ 8) character rows are actually available. Such a program would display a "squashed" screen image because it neglects to display the seven rows of characters.

You can remedy this situation by running the VGARES.C program as a DOS command (from the command line or a batch file) before running an EGA-specific application. The program is written in C for the Microsoft compiler version 4 or later. If it finds a valid resolution value on its command line, it calls the C library function **int86()** to execute software interrupt 10H, function 12H, with BL = 30H. This sets the resolution bits in the BIOS flag byte at address 40:89H according to the value in AL: 200 lines for AL = 0, 350 for AL = 1, 400 for AL = 2. The new vertical resolution takes effect the next time the video mode is set with interrupt 10H, function 0, so the program determines the current mode and then sets it to the same value. Whether the resolution flags are changed or not, the program inspects the flag byte and displays the resolution corresponding to the settings of bits 4 and 7; see the second **switch** statement in listing 1.

2 BACKGROUND COLORS IN GRAPHICS MODES

Displaying text in one of the all-points-addressable (APA) graphics modes is not easy—each pixel in a character's bit map must be turned on individually. Fortunately, the PC's ROM BIOS provides routines to automate this process, so that, to a program, displaying a character is equally easy in either text or graphics modes. However, several significant differences exist.

In text mode, the program has complete control over both the foreground and background colors of the characters. But in graphics mode, only the foreground color can be specified; for the background, the programmer normally has the same choice as the purchaser of a Model T—black.

The BIOS character-drawing routines are called by interrupt 10H, function 9. Register AL contains the ASCII code of the character to be written, CX the number of times to repeat the character, and BL the video attribute. In text modes, the low-order nibble of the attribute byte specifies the foreground color, the high-order nibble the back-

ground color and blinking. In graphics modes, the low-order nibble still sets the foreground color, or the color of each pixel within the character's bit map. But the background color cannot be specified directly. If the high-order bit of the attribute byte is zero, the background is set to black; if the bit is 1, the character's bit map is combined by an exclusive-or operation with the existing contents of video memory. The result is that the background color is unchanged, the foreground color becomes the exclusive-or of the numeric values of the background color and the low-order nibble of BL.

For example, suppose that in EGA graphics mode 10H (640 by 350 pixels by 16 colors) the screen is solid green, color value 2. Writing text with an attribute of 5 results in magenta characters on a black background, but an attribute of 85H produces white characters on a green background. The resulting foreground color value is 5 XOR 2 = 7 (white); however, the background color remains unchanged.

Given an existing background color x and a desired foreground color y to superimpose upon it, the attribute is calculated as (x XOR y) + 80H. For example, to overlay yellow (color 14) on an existing magenta (color 5) background, the attribute is 8BH (14 XOR 5 = 11 decimal).

This provides a convenient method of writing to areas that have been colored by previous graphics operations, but the desired background color is not always available. If the objective merely is to produce a row of character cells in the background color, it is both inconvenient and time consuming to calculate the pixel coordinates of the required rectangle and turn on every pixel within it. The easiest and fastest approach is to write a string of the character with the extended ASCII code DBH, which is a solid block with every pixel in a cell turned on (the inverse of a space), then to overlay the desired characters into the same character cells.

This brings up another complication—on CGA and many EGA systems, the upper half of the extended ASCII character set is undefined in graphics modes 4, 5, and 6. It is defined in all APA modes on VGA systems and in EGA modes DH through 10H. Also, many clone EGA boards extend the IBM standard by defining the upper half of the character set, even in modes 4, 5, and 6. Although all EGA boards must provide 8-by-8-pixel font tables for the entire character set, those that adhere strictly to the IBM standard do not automatically provide a pointer to the upper half of this table; for them, an easy fix is available.

In graphics modes, the BIOS video routines look for the pointer to the upper half of the font tables in the vector for interrupt 1FH. If the segment portion of this vector is zero, the high-bit characters are undefined. Setting the vector to the font table in the EGA ROM defines them. This address can be obtained by executing interrupt 10H, function 11H, with AL = 30H and BH = 4. Function 25H of interrupt 21H can then set the interrupt vector to this address, as is illustrated by the following sequence of instructions.

```
MOV  AX,1130
MOV  BH,04
INT  10
MOV  DX,BP
MOV  AX,ES
MOV  DS,AX
MOV  AX,251F
INT  21
INT  20
```

The INT 10H call returns the address in ES:BX, so it must be moved to DS:DX for the INT 21H call. This program is more conveniently assembled directly to a .COM file with a debugger than with an assembler, linker and EXE2BIN. Note that the INT 10H, function 11H call is supported only on EGA and VGA systems, not on the CGA because the latter does not have font tables in addressable ROM space. The program described next will indicate whether the above is necessary, but it does not hurt to run it in any case.

The COLORTXT.PAS program (listing 2) demonstrates the writing of text on various colored backgrounds in EGA graphics modes. It was first written in Turbo Pascal 3.02A, then converted to version 4 with minimum changes. It will run only on systems with EGA or VGA video controllers.

The main procedure of COLORTXT.PAS saves the current video mode, switches into mode 10H, displays a screenful of colorful text, then waits for a keystroke. Upon receiving it, it switches to mode 6 (640 by 200 monochrome), displays a line of white-on-black text and a line of black-on-white, and again pauses for a keystroke. Then it restores the original video mode and exits.

If on the black-and-white screen the second line is the same as the first, then the upper half of the character set is undefined. Run the program given above, then rerun COLORTXT.PAS.

COLORTXT.PAS includes two procedures that perform low-level video BIOS functions. **Set_Cursor_Position** positions the cursor with video BIOS function 2. **Write_Byte_Attribute** writes a character with its associated attribute byte to the display, using video BIOS function 9.

Procedure **Screen_Write** is the key. It displays a character string in any one of 16 foreground colors superimposed on a background of any one of 16 colors—this includes white-on-black or black-on-white in any of the monochrome graphics modes. The parameters specify the desired final colors; the procedure calculates the XORed value passed to BIOS video function 9. As written, the procedure is useful only for modes 10H and 6 and needs changes to work with CGA mode 6 or VGA mode 13H. The former permits 4 colors and the latter 256, so the calculation of the foreground attribute must be adjusted accordingly.

LISTING 1: VGARES.C

```
/*********************************************************************
 *
 * Name:        vgares.c
 *
 * Function:    select vertical resolution in VGA alphanumeric modes
 *
 * Syntax:      VGARES [vertical resolution]
 *
 * Notes:       Specified vertical resolution must be 200, 350, or 400
 *               scan lines.  If no resolution is specified, the
 *               program displays the current vertical resolution.
 *
 *********************************************************************/

#include <dos.h>

main( argc, argv )
int     argc;
char    **argv;
{
    struct WORDREGS     regs;           /* used by int86() */
    unsigned char far   *FLAGS = (unsigned char far *)0x00400089;
    int                 ScanLines;

    /* set vertical resolution if specified in command line */
```

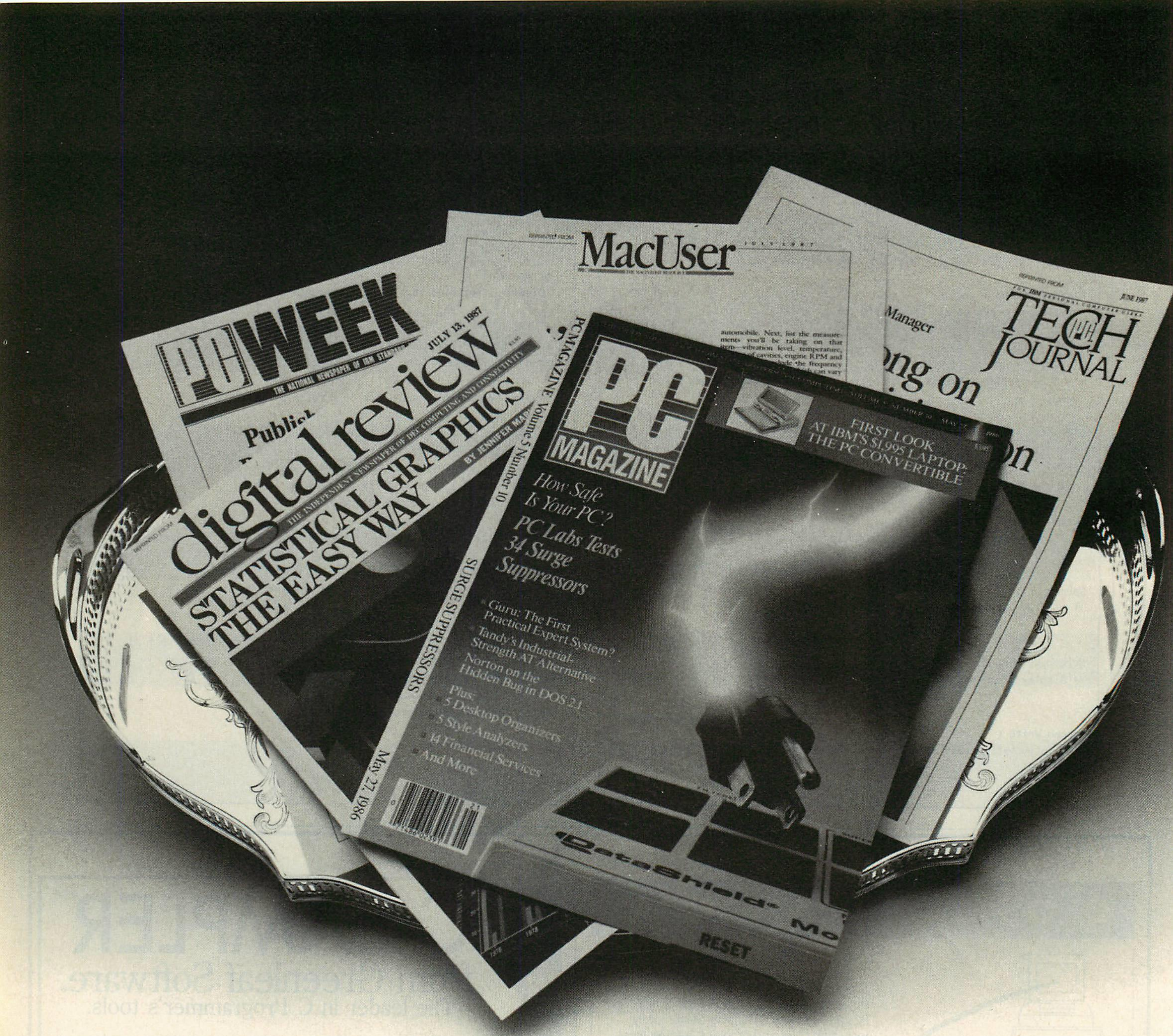
```
  if( argc == 2 )
  {
    sscanf( argv[1], "%d", &ScanLines );

   /* validate scan lines arg & set up for INT 10H func 12H */

    switch( ScanLines )
    {
      case 200:
        regs.ax = 0x1200;
        break;

      case 350:
        regs.ax = 0x1201;
        break;

      case 400:
        regs.ax = 0x1202;
        break;

      default:
        printf( "\nError: Scan lines value should be");
        printf( " 200, 350, or 400\n" );
        exit( 1 );
    }                           /* end of switch */

    /* bl = 30h, Select scan lines */
      regs.bx = 0x30;
      int86( 0x10, &regs, &regs );

      /* set scan lines by setting video mode to current value */
      regs.ax = 0x0F00;           /* Get Video Mode */
      int86( 0x10, &regs, &regs );

      regs.ax &= 0x00FF;          /* Set it to same mode (in AL) */
      int86( 0x10, &regs, &regs );
    }                             /* endif argc == 2 */

    /* show current vertical resolution */
    /* test bits 4 and 7 of BIOS flags byte  @ addr 40:89 */
    switch( *FLAGS & 0x90 )
    {
      case 0x00:
        ScanLines = 350;
        break;

      case 0x10:
        ScanLines = 400;
        break;

      case 0x80:
        ScanLines = 200;
        break;
    }

    printf( "\nCurrent vertical resolution (scan lines) is %d\n",
      ScanLines );

    exit( 0 );
}
```

LISTING 2: COLORTXT.PAS

```
{$R-}     {Range checking off}
{$B+}     {Boolean complete evaluation on}
{$S+}     {Stack checking on}
{$I+}     {I/O checking on}
{$N-}     {No numeric coprocessor}
{$M 65500,16384,655360} {Turbo 3 default stack and heap}

{_____________________________________________________________

COLORTXT.PAS shows how to display character strings in all
combinations of foreground and background colors for EGA mode 10h,
640x350 pixels with 16 colors, then in normal and inverse video for
CGA mode 6, 640x200 pixels monochrome.  This method works for all
16-color EGA and VGA modes, and for CGA, EGA, VGA, and MCGA
monochrome modes.  For CGA mode 5, 320x200 pixels and 4 colors,
the concept remains the same, but you may have to make changes in
the Screen_Write routine.  VGA/MCGA mode 13h is another topic.

This version of COLORTXT is written for Turbo Pascal Release 4.0.

WARNING - To run as written, your PC must have either an EGA or
          a VGA video controller.

Copyright, 1987, Ben Myers
 _____________________________________________________________
}

program colortxt;

Uses
  Crt, Dos, Turbo3;

const
  _Copyright : string[80] = 'Copyright 1987, Ben Myers';
  _VIDEO = $10;      { BIOS video interrupt }

var
   __BIOSReg : Registers;
   Previous_Mode : byte;
   TempCh : char;
   Display_String : string[80];
   i, k, Color_Index : integer;

{____________________________________________________________

Set_Cursor_Position - Position the cursor on active display page zero.

X and Y are the row and column to position the cursor, relative to 1,
like Turbo GoToXY.  This procedure gives correct results for values
of X and Y within the range allowed by the current video BIOS mode.
 ____________________________________________________________
}

procedure Set_Cursor_Position ( X, Y : integer );

begin {Set_Cursor_Position}

  with __BIOSReg do
    begin
      Ah := $02;  { BIOS VIDEO subfunction 2, Set Cursor Position. }
      Bh := 0;              { Video display page zero }
      Dh := pred(Y);        { Make zero relative }
      Dl := pred(X) and $FF;
    end;
    Intr(_VIDEO,Dos.Registers(__BIOSReg));
{! 1. Parameter to Intr must be of the type Registers defined
 in DOS unit.}
end; {Set_Cursor_Position}

{_____________________________________________________________

Write_Byte_Attribute - Write copies of a character with a specified
                       attribute beginning at current cursor
                       position.

Data_Ch     The character to display
Count       The number of copies of Data_Ch to display
Fore, Back  The foreground and background attributes for the character.
 _____________________________________________________________
}

procedure Write_Byte_Attribute (Data_Ch : char;
                                Count, Fore, Back : integer);

begin {Write_Byte_Attribute}
   if (Count > 0) then
   with __BIOSReg do
    begin
     Ah := $09;  { BIOS video subfunction 9, }
     { Write Attribute/Character at Current Cursor Position }
     Al  := Ord(Data_Ch);
     { Use video display page zero (Bh), and          }
     { force reasonable values for the attributes. }
     Bx  := ((Back and $0F) shl 4) or (Fore and $0F);
     Cx  := Count;
     Intr(_VIDEO,Dos.Registers(__BIOSReg));
```

```
    end;
end; {Write_Byte_Attribute}

{_____________________________________________________________

Screen_Write - Display a character string in color on a background
               color.

Phrase              The character string to be displayed
Text_Row, Text_Col  The starting row and column on which to display
FColor, BColor      The foreground and background colors to use

______________________________________________________________
}

Procedure Screen_Write ( Phrase : String; Text_Row, Text_Col,
                         Fcolor, Bcolor: integer );
const
{ Font character consisting of all bits (a little square) }
    White_Space : char =  #$DB;

var
  j : integer;
  Phrase_Size : integer;
  Effective_Foreground_Color : byte;

begin  { Screen_Write }

  Phrase_Size := Length ( Phrase );
  if Bcolor <> Black then
    begin  { Display string against non-black background }
      { Paint Phrase_Size bytes of all bits in background color }
      Set_Cursor_Position ( Text_Col, Text_Row );
      Write_Byte_Attribute ( White_Space, Phrase_Size,
                             BColor, Black );
      Effective_Foreground_Color := FColor xor BColor;
      { For EGA, VGA, ATs, XT/286's, and PS/2's, the for statement
        could be replaced by a single call to video BIOS function
        $13, Write String.
      }
      for j := 1 to Phrase_Size do
      if Phrase [j] <> ' ' then
        begin
          Set_Cursor_Position ( Text_Col + pred(j), Text_Row );
          Write_Byte_Attribute ( Phrase [j], 1,
                                 Effective_Foreground_Color,
                                 8 {xor font bits on background});
        end;
    end   { Display string against non-black background }
else
    begin  { Display string against black background }
      { Write the character string }
      for j := 1 to Phrase_Size do
      if Phrase [j] <> ' ' then
        begin
          Set_Cursor_Position ( Text_Col + pred(j), Text_Row );
          Write_Byte_Attribute ( Phrase [j], 1, FColor, Black );
        end;
    end;  { Display string against black background }
end;  { Screen_Write }

{_____________________________________________________________

Main Procedure

______________________________________________________________
}
begin  {colortxt}

{ Display 16 colors 640x350 with EGA mode 10h }

  { BIOS VIDEO subfunction 0Fh, Read Current Video State. }
  __BIOSReg.Ah := $0F;
  Intr(_VIDEO,Dos.Registers(__BIOSReg));
  Previous_Mode := __BIOSReg.Al;
  { BIOS VIDEO subfunction 0, Set Mode. }
  __BIOSReg.Ax := $0010;
```

```
Intr(_VIDEO,Dos.Registers(__BIOSReg));
Display_String := ' Display all colors on line 1. ' + _Copyright;
for i := 1 to length(Display_String) do
  begin
    { Don't display a black character }
    Color_Index := (i mod 15) + 1;
    Screen_Write ( Display_String[i], 1, i, Color_Index, Black );
  end;

Display_String :=
  ' Display black on all background colors on line 2. ';
for i := 1 to length(Display_String) do
  begin
    Color_Index := (i mod 15) + 1;
    Screen_Write ( Display_String[i], 2, i, Black, Color_Index );
  end;

{ Display all possible combinations }
for k := 0 to 15 do
for i := 0 to 15 do
  Screen_Write ( chr(ord('A')+k+i), 3+k, 1+i, i, k);

Screen_Write (
 ' Display blue text on light cyan background on line 19. ',
               19, 1, Blue, LightCyan );

Screen_Write (
 ' Display yellow text on red background on line 20. ',
               20, 1, Yellow, Red );

Screen_Write (
 ' Display green text on yellow background on line 21. ',
               21, 1, Green, Yellow );

Screen_Write (
 ' Display yellow text on green background on line 22. ',
               22, 1, Yellow, Green );

Screen_Write (
 ' Display magenta text on light green background on line 23. ',
               23, 1, Magenta, LightGreen );

Screen_Write (
 ' Display light cyan text on blue background on line 24. ',
               24, 1, LightCyan, Blue );

Screen_Write (
 ' Display light blue text on light magenta background on line 25. ',
               25, 1, LightBlue, LightMagenta );

while not KeyPressed do;   { Wait for a keystroke }
Read ( kbd, TempCh );         { Now get rid of it }

{ Display monochrome in CGA mode 6, 2 color 640x200  }

{ BIOS VIDEO subfunction 0, Set Mode. }
__BIOSReg.Ax := $0006;
Intr(_VIDEO,Dos.Registers(__BIOSReg));
Screen_Write ( ' Display white on black on line 1. ',
               1, 1, White, Black );

Screen_Write ( ' Display black on white on line 3. ',
               3, 1, Black, White );

while not KeyPressed do;   { Wait for a keystroke }
Read ( kbd, TempCh );         { Now get rid of it }

{ Leave the video mode as it was found when program started up }
{ However, active display page from that mode is not restored  }
{ BIOS VIDEO subfunction 0, Set Mode. }
__BIOSReg.Ax := Previous_Mode;
Intr(_VIDEO,Dos.Registers(__BIOSReg));

end.   {colortxt}
```



Seconds Never Looked So Good!

Attract potential customers, distributors, and vendors with our tastefully designed reprints. Revitalize your article with front page coverage from one of the world's leading computer magazines.

Ziff-Davis reprints are concisely reformatted on 80 pound coated paper and printed from full color to black and white.*

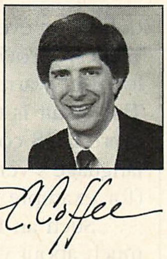
To find out how you can have your article reprinted, call or write today.

Jennifer Locke-Reprints Manager, Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016, (212) 503-5447.

ZD

*minimum order 500 reprints.

OUTFITTING THE END USER



Environmental Cover-ups

When building environments, the tendency is to cover up the seamy underside of the computer. But, perhaps honesty is the best policy.

Do you remember the first time that you *used* a desktop computer? I don't mean the first time you ran a word processor or worked with a spreadsheet, but rather the first time you had to format a disk, for example. I mean those moments when you are not doing anything useful; you are just using a computer. Ugly, isn't it?

A lot of what we all do is directed toward making computers more accessible to people who are intimidated by them. We can do it either by making the computer less visible or by taking the time to make computers less mysterious. I strongly favor the second approach as a long-range goal, but the first one usually gets new users productive much more quickly.

BEYOND BATCH FILES

For many of us, the ancient and honorable menu—whose choices are merely the names of batch files—is an effective solution that isolates users from the perils of the command line. The menu is especially useful with a hard disk, which long ago crossed the line from being just a power-user accessory to being recognized as essential to casual-user support. No more formatting. No more labels written with (shudder) ballpoint pen.

Most batch menus present their options in terms of *applications*; they have an entry for the word processor, an entry for the spreadsheet, and so on. In order to build an environment that has real impact on the usefulness of the machine, however, menu entries should focus instead on *tasks*, exploiting application features to automate entire sessions rather than just launching a general-purpose program.

Take the case of a user whose only interaction with Lotus 1-2-3 is to use a spreadsheet template that records purchases of advertising space. Within the spreadsheet are macros for all the basic operations, including a quit macro that saves the file and returns to DOS, where the top-level batch file takes over and restores the menu. However, this user still had to deal with the mechanics of getting into 1-2-3 and loading the file, which clearly aggravated her. "Why can't I just have an Ad Space choice on my menu?" she asked.

Even this is possible, because 1-2-3 (at least, Release 1A) automatically loads any worksheet in its default directory that is named AUTO123. A batch file can copy the ADSPACE file into the 1-2-3 directory as AUTO123, run 1-2-3, and on exit copy AUTO123 back into ADSPACE. This has the added advantage of leaving the old file intact if anything goes wrong during the session and forces an abnormal exit. The user now has the simple menu entry, Ad Space: ADS{enter}. The next screen the user sees is the spreadsheet's help panel, listing the macro keystrokes, courtesy of an auto-executing macro.

If you do not need automatic backup and/or your files are too big to copy conveniently, just rename the target file instead of copying. For that matter, most programs support the specification of a target file as an option on the command line; this makes the idea of a task-oriented batch file that much easier to use. Crosstalk XVI scripts are an excellent example: one command from the DOS prompt and the user is logged in and being prompted for the password.

A .COM file that triggers a reboot can make batch files even more sensitive to task requirements. With this mechanism, a batch file can copy task-specific versions of CONFIG.SYS and AUTOEXEC.BAT into place, then simply reboot the machine. RAM-disk allocation, background utilities, and so on can be optimized for each task. This is especially convenient if you want to use a software-configurable memory board as extended-memory RAM disk for some applications—such as compiling—but as expanded memory for packages that can better use that mode.

Finally, for advanced requirements, you might consider developing your own shell program: a simple loop, input/test/EXEC/repeat, where the *test* portion is anything from a string matcher to a natural language interpreter (as might be written in Turbo Prolog). The shell can allow as many synonyms for a program or task as you like, without each one taking up the large minimum space that DOS re-

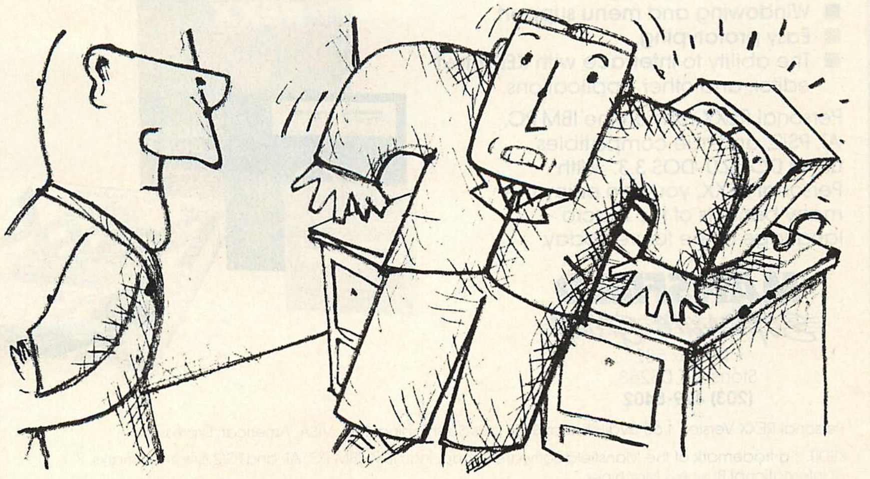


ILLUSTRATION • MACIEK ALBRECHT

quires for even the smallest batch file. You can provide interactive subroutines for user warning or input clarification. The result is the issuance of one or more DOS commands through your language's version of the EXEC facility (for example, SHELL in BASIC).

Such an interface will know as much about your work routine as you care to tell it—just leave enough memory to run useful applications as well.

WINDOWS AND WORKLOAD

Where more integration between applications is desired, I have had excellent experience with Quarterdeck's DESQview. On power-up, one such system of mine immediately opens three windows: one for direct access to DOS, one for the local area network's electronic mail utility (automatically executing a Retrieve Mail command), and a text edit window that retrieves and displays a "to do" list. Other tasks and applications are built into the extendible DESQview menu. Any user who can find the Alt key can use a DESQview-based environment.

DESQview's macro feature lets menu entries be even more task-oriented. My favorite is a macro that opens WordPerfect, gives it a command to display the last memo number used, then opens a New Memo template file and positions the cursor to type the next sequential number. (I never did figure out a way to make it increment by itself, but that does not mean it can't be done.) Then the macro takes over again; it backs up three spaces, copies the number just typed, and edits the header accordingly so that pages 2 and beyond will be properly labeled. It then places the cursor at the Title position and returns control to the user.

This is genuine workload reduction: decide to type a memo, give one command, type the memo number, and go on. This illustrates the value of a fully programmable environment, as opposed to an enticing but dumb graphics interface that only makes it incrementally easier for you to continue doing all the work yourself.

PROGRAMMABLE ENVIRONMENTS

My main application on PC-class machines today is Ashton-Tate's Framework II. Its integrated outlining helps me think about a project from the beginning; once the various outline entries have resolved themselves into documents, spreadsheets, and databases, it lets me move information easily and naturally between different frames. I am also thoroughly spoiled by the way each frame remembers where I was the last time I worked inside it—like having place markers for every panel of a spreadsheet and every section of a document.

The communications facility is surprisingly good, with full macro capabilities. Communications remain the most obscure applications for many users, so automating routine sessions should be a high-priority item. Framework transcribes a session into the system as a document, in a frame of its own; the user can browse, or even edit it, while new material is still coming in at the bottom. Now, *that's* integration.

Framework's integrated programming language, known as FRED, gives it some of the best features of both DESQview and the personal shell alternative that is described above. At the lowest level, FRED lets you comment on the formulas in your spreadsheets; at the highest level, it makes the whole environment programmable.

As we climb this ladder of increasingly advanced environments—menus that provide a superficial improvement in ease of access, windowing systems that add a superstructure of actual integration, and, finally, fully programmable toolkits such as Framework—several themes emerge.

First, users would rather think about tasks than applications. Menu choices called New Memo or Ad Space correspond to real tasks, whereas Word Processor is just an intermediate step on the way to doing useful work. Your own work may be interrupt-driven, but do not let that blind you to the opportunities for helping other people with more predictable jobs.

Second, users expect consistency. This is supposed to be the motivation for Microsoft Windows, but the transition is going to be painful. Today, for example, running "non-aware" applications under Windows means keeping track of two cursors (one that is mouse driven for Windows navigation and another that is keyboard driven for the application). I am constantly positioning the wrong cursor, then realizing that it will not do what I want. Inconsistency diverts the user's attention from task to tool. Minimize it in whatever aids you provide.

Third, whatever you *want* your users to do should be *easy* for them to do. This is the lesson of Framework, which makes it just as easy to dial up the mainframe and get the real num-

bers—right now, while you are wondering what values to use in a spreadsheet under construction—as it is to concoct an estimate. By contrast, an ordinary spreadsheet (no matter how powerful it may be) makes it easier to estimate data than to go out and get the real thing. Can you blame users for following the lead that their environment provides?

That last question brings up some fundamental issues of what we want our systems to do. You may have to make some basic decisions in the coming year that could influence the whole direction of your organization's computing; a little thought today may make a big difference in what you spend and what you can do for the next five years or so. What are the options?

THE MASQUERADE

You can follow one of two paths in applications design. The first, which dominates productivity applications today, is the masquerade approach, in which computers look more and more like the special-purpose machines that they emulate. This is a seductive path, because the next step usually calls attention to itself through user errors or complaints; you always feel as if you are closing in on the final solution.

The canonical example of this approach is the early edition of 1-2-3, with its Disk Manager and File Manager utilities. Some users had no idea of what to do at a DOS prompt, because their AUTOEXEC took them into 1-2-3 where they stayed all day.

The undoing of the masquerade is that try as you may, you cannot conceal forever the fact that the dumb box under all those layers of makeup knows nothing about your assorted words and numbers. It only understands bytes. The more successful you are in concealing the byte level, however, the more complete is the user's helplessness upon falling off the edge of your imitation world.

Whatever happened to the vision of the early 1950s? By this time, we were to have computers that eliminated the need for human attention to routine tasks, instead of merely speeding up our least interesting activities and creating whole new categories of drudgery in the process.

The success of the emulation approach to applications design may be linked to this lack of real progress in the *kind* of help that computers provide. Have you looked at what your power users are doing lately? Thanks to the wonders of laser-printed WYSIWYG, they are worrying about output placement on the page in units of three-hundredths of an inch! The increasing emphasis on output appearance increases the time needed to produce the answer; worse still, it diverts attention from the issue of what the question ought to be.

This is the dark side of environmental impact. A strong sense of purpose is necessary to avoid thinking that the tasks your tools do best must be important tasks. It is often said, "If your only tool is a hammer, every problem looks like a nail." The real problem is that if your only tool is a hammer, banging on anything looks productive.

You may have to make some basic decisions in the coming year that could influence the direction of your organization's computing.

This kind of mass hypnosis has occurred before on an even larger scale. As John Gall observes in his minor classic *Systemantics*, "The reason we think the auto industry is meeting our needs is that we have almost completely forgotten what we originally wanted, namely, a means of going from one place to another that would be cheap, easy, convenient, safe, and fast. We have been brainwashed into thinking that the Detroit product meets these requirements." The masquerade approach to applications design—and, for that matter, the emphasis on LANs rather than multiuser systems—may be going down the same, expensive, ultimately counterproductive path.

THE HONESTY OPTION

Imagine that you are trying to get a view of the world. Trying to see the whole thing, you go higher and higher. If you get infinitely high, you will be able to see—what? Only half of it. This is the trap of the emulation approach to applications design: even with infinite effort, the designer still must conceal part of the system's capability to disguise its true nature.

Contrast this with the person who goes to the center of the sphere. Once inside, he can look around and see the entire world. This is the perspective offered by UNIX and similar data-stream environments. The user knows the machine is pumping around streams of characters and can readily exploit the fact. Many UNIX applications work in the manner of a filter, allowing input/output redirection to link them and produce impressive hybrid capability. Users are much less often surprised, because the function of each small tool is much better defined than are the Swiss Army knives that people try to build under DOS.

"But UNIX is just too cryptic for the average user," some will say. To this I reply, "Don't sell your users short." I have seen so-called average users with high school educations sneer at well-known WYSIWYG word processors; why should they go to the trouble of making it look right line by line, when they can just tell **troff** (a UNIX text formatting utility) what they want and let it take care of the details?

This is not to say interactivity is not important, but the emphasis must be on interactivity *directed to some end*. I would rather use old-fashioned, menu choices to get precisely the answer I need to the same complex question that I ask every day than have all the benefits of a graphics interface to help me do the work by hand.

THE HITCHHIKER MODEL

Douglas Adams offers a tongue-in-cheek model of societal evolution in *The Hitchhiker's Guide to the Galaxy*. For those of you who missed it, Adams asserts (not too seriously) that every civilization passes through three stages: survival, inquiry, and sophistication. These are represented, he adds, by three questions: How can we eat? Why do we eat? Where shall we have lunch?

In many organizations, computing has passed the survival stage (Quick, what's the answer?); has passed the inquiry stage (What other questions should we be asking?); and has now reached the dubious summit of sophistication (Why are you doing this chart in 1-2-3? Let me show you this new graphics package . . .). The number of people whose best creative energy goes into "power using" may be a far more costly problem than many organizations are prepared to admit.

We should consider the state of computing on this somewhat philosophical level because of an unusual pair of transitions in progress today. We are going through a wave of increased corporate confidence in desk-

top computing with attendant plans for dramatic expansion, and at the same time we are reaching an equally important threshold in the available technology—specifically, the combination of the 80386 processor on the hardware side and OS/2 or UNIX (XENIX)/386 on the systems software side. These transitions make it both possible and essential for you to make some expensive decisions, in the way that is most effective *for your organization*, over the next year or so.

What I am actually talking about is the answer to the question: "What do we do with all the old PCs?" A lot of 8088- and 8086-based machines are out there. You can say, "They meet the need," perhaps by adding expanded memory and bigger, faster hard disks for improved performance under DOS or upgrading their processors, memory, and displays for OS/2.

Or you can say, "Look. We really do plan to have some kind of information appliance on the desk of every employee by 1992. Network cost per user drops—quickly—as the number of users goes up." Then you start to look at XENIX on a 386 for a small department or IX/370 on a 9370 for a larger group, and let your PCs grow old gracefully as terminal emulators with useful local storage and the ability to do off-line editing.

The choice between multiuser options and the OS/2 Presentation Manager is mutually exclusive, unless you want to tie yourself to IBM's Systems Application Architecture. Most of the multiuser systems drive graphics displays only on the primary console. If multiuser is the way you want to go, investing in the widely distributed processing and the graphics display horsepower that the Presentation Manager requires may be just a waste of money.

In fact, DOS emulations under UNIX/XENIX on the 386 may prove themselves to be even more useful than the OS/2 compatibility box—they will run multiple "old" applications at once. The availability of native UNIX and XENIX versions of DOS applications also continues to improve (for example, Innovative Software's Smart family and Microsoft's XENIX Word). The various UNIX/XENIX shell programs offer considerable facilities for programming routine sessions, and one (SCO's MultiView) even claims to provide DESQview-like multiwindow cut-and-paste integration between concurrent tasks—not just on the console, but on multiple terminals as well.

In the end, you must evaluate the importance of information sharing. Figuring out what your actual problems are, and what they require in information production and management, is harder than going with the newest generation of stand-alone, general-purpose tools and counting on users to know what to do with them. The problem is that today's most enticing tools send the wrong message to users about which jobs are most important.

Planning a path of systems development that balances the degree of sharing with the power of the individual nodes in the network is the key strategic responsibility for information professionals today. Next month, I will look at some techniques that can help to clarify this process.

Peter C. Coffee is managing partner of SolveWare, a developer and business computing consultant, and is active in AI and distributed computing applications for aerospace and educational clients.

NEED IT FAST? WANT TO KNOW MORE? DON'T HAVE TIME TO SHOP AROUND?

INTRODUCING...

If you're like most system professionals you're up-to-date about the products in the PC marketplace. You're aware of brand and model differences, are informed about connectivity and compatibility problems, and you shop for competitive prices and fast service.

You're also probably among the many PC TECH JOURNAL readers who purchase by mail. That's why we're starting THE MART—PC TECH JOURNAL'S First Class Mail-Order Section.

Starting this month, and every month hereafter, you'll find the products you're looking for advertised in THE MART—and you'll benefit from the fast service and helpful support that identifies PC TECH JOURNAL advertisers.

If you're ready for First Class service, you're ready for THE MART.

UNLEASH YOUR 80386!

Your 80386-based PC should run two to three times as fast as your old AT. This speed-up is primarily due to the doubling of the clock speed from 8 to 16 MHz. The new MicroWay products discussed below take advantage of the real power of your 80386, which is actually 4 to 16 times that of the old AT! These new products take advantage of the 32 bit registers and data bus of the 80386 and the Weitek 1167 numeric coprocessor chip set. They include a family of MicroWay 80386 compilers that run in protected mode and numeric coprocessor cards that utilize the Weitek technology.

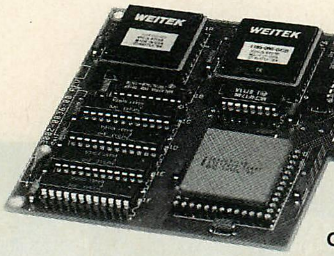
The benefits of our new technologies include:

- An increase in addressable memory from 640K to 4 gigabytes using MS-DOS or Unix.
- A 12 fold increase in the speed of 32 bit integer arithmetic.
- A 4 to 16 fold increase in floating point speed over the 80387/80287 numeric coprocessors.

Equally important, whichever MicroWay product you choose, you can be assured of the same excellent pre- and post-sales support that has made MicroWay the world leader in PC numerics and high performance PC upgrades. For more information, please call the Technical Support Department at

617-746-7341

After July 1988 call 508-746-7341



mW1167 Numeric Coprocessor Board

MicroWay® 80386 Support

MicroWay 80386 Compilers

NDP Fortran-386 and **NDP C-386** are globally optimizing 80386 native code compilers that support a number of Numeric Data Processors, including the 80287, 80387 and mW1167. They generate mainframe quality optimized code and are syntactically and operationally compatible to the Berkeley 4.2 Unix f77 and PCC compilers. MS-DOS specific extensions have been added where necessary to make it easy to port programs written with Microsoft C or Fortran and R/M Fortran.

The compilers are presently available in two formats: Microport Unix 5.3 or MS-DOS as extended by the Phar Lap Tools. MicroWay will port them to other 80386 operating systems such as OS/2 as the need arises and as 80386 versions become available.

The key to addressing more than 640 kbytes is the use of 32-bit integers to address arrays. NDP Fortran-386 generates 32-bit code which executes 3 to 8 times faster than the current generation of 16-bit compilers. There are three elements each of which contributes a factor of 2 to this speed increase: very efficient use of 80386 registers to store 32-bit entities, the use of inline 32-bit arithmetic instead of library calls, and a doubling in the effective utilization of the system data bus.

An example of the benefit of excellent code is a 32-bit matrix multiply. In this benchmark an NDP Fortran-386 program is run against the same program compiled with a 16-bit Fortran. Both programs were run on the same 80386 system. However, the 32-bit code ran 7.5 times faster than the 16-bit code, and 58.5 times faster than the 16-bit code executing on an IBM PC.

NDP FORTRAN-386™ $595
NDP C-386™ $595

MicroWay Numerics

The **mW1167™** is a MicroWay designed high speed numeric coprocessor that works with the 80386. It plugs into a 121 pin "Weitek" socket that is actually a super set of the 80387. This socket is available on a number of motherboards and accelerators including the AT&T 6386, Tandy 4000, Compaq 386/20, Hewlett Packard RS/20 and MicroWay Number Smasher 386. It combines the 64-bit Weitek 1163/64 floating point multiplier/adder with a Weitek/Intel designed "glue chip". The mW1167™ runs at 3.6 MegaWhetstones (compiled with NDP Fortran-386) which is a factor of 16 faster than an AT and 2 to 4 times faster than an 80387.

mW1167 16 MHz $1495
mW1167 20 MHz $1995

Monoputer™ - The INMOS T800-20 Transputer is a 32-bit computer on a chip that features a built-in floating point coprocessor. The T800 can be used to build arbitrarily large parallel processing machines. The Monoputer comes with either the 20 MHz T800 or the T414 (a T800 without the NDP) and includes 2 megabytes of processor memory. Transputer language support from MicroWay includes Occam, C, Fortran, Pascal and Prolog.

Monoputer T414-20 with 2 meg[1] ... $1495
Monoputer T800-20 with 2 meg[1] ... $1995

Quadputer™ can be purchased with 2, 3 or 4 transputers each of which has 1 or 4 megabytes of memory. Quadputers can be cabled together to build arbitrarily fast parallel processing systems that are as fast or faster than today's mainframes. A single T800 is as fast as an 80386/mW1167 combination!

Biputer™ T800/T414 with 2 meg[1] $3495
Quadputer 4 T414-20 with 4 meg[1] ... $6000

[1] Includes Occam

80386 Multi-User Solutions

AT8™ - This intelligent serial controller series is designed to handle 4 to 16 users in a Xenix or Unix environment with as little as 3% degradation in speed. It has been tested and approved by Compaq, Intel, NCR, Zenith, and the Department of Defense for use in high performance 80286 and 80386 Xenix or Unix based multi-user systems.

AT4 - 4 users $795
AT8 - 8 users $995
AT16 - 16 users $1295

Phar Lap™ created the first tools that make it possible to develop 80386 applications which run under MS-DOS yet take advantage of the full power of the 80386. These include an 80386 monitor/loader that runs the 80386 in protected linear address mode, an assembler, linker and debugger. These tools are required for the MS-DOS version of the MicroWay NDP Compilers.

Phar Lap Tools $495

PC/AT ACCELERATORS

287Turbo-10 10 MHz $450
287Turbo-12 12 MHz $550
287TurboPlus-12 12 MHz $629
FASTCACHE-286 9 MHz $299
FASTCACHE-286 12 MHz $399
SUPERCACHE-286 .... $499

MATH COPROCESSORS

80387-20 20 MHz $895
80387-16 16 MHz $495
80287-10 10 MHz $349
80287-8 8 MHz $259
80287-6 6 MHz $179
8087-2 8 MHz $154
8087 5 MHz $99

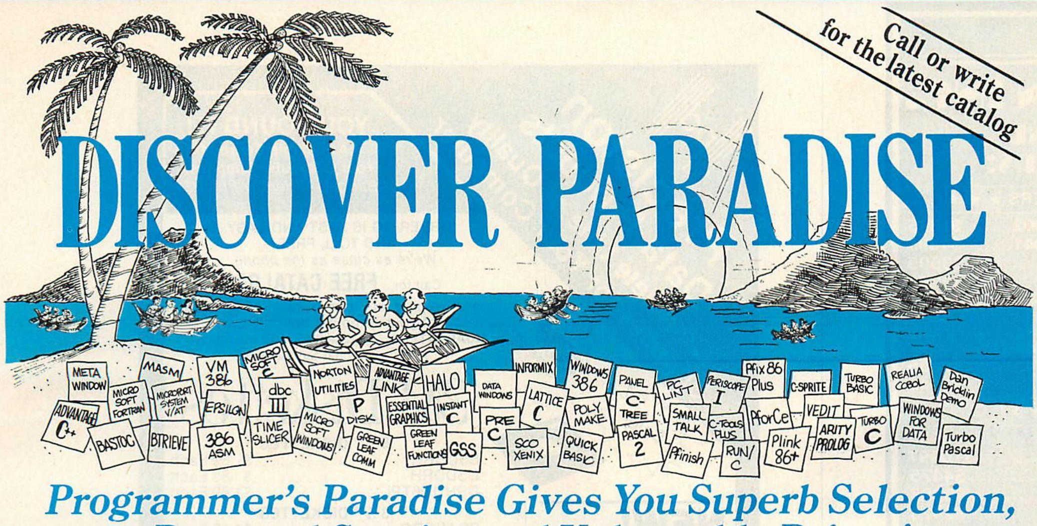
Micro Way

The World Leader in PC Numerics

P.O. Box 79, Kingston, Mass. 02364 USA (617) 746-7341
32 High St., Kingston-Upon-Thames, U.K., 01-541-5466
St. Leonards, NSW, Australia 02-439-8400

Call or write for the latest catalog

DISCOVER PARADISE



Programmer's Paradise Gives You Superb Selection, Personal Service and Unbeatable Prices!

Welcome to Paradise. The microcomputer software source that caters to your programming needs.
Discover the Many Advantages of Paradise…

- Lowest price guaranteed
- Latest versions
- Huge inventory, immediate shipment
- Knowledgeable sales staff
- Special orders
- 30-day money-back guarantee

Over 500 brand-name products in stock—if you don't see it, call!

We'll Match Any Nationally Advertised Price.

386 SOFTWARE

	LIST	OURS
ADVANTAGE 386 C OR PASCAL	895	799
MICROPORT SYSTEM V/386 (COMPLETE)	799	679
MS WINDOWS/386 *SPECIAL, NEW*	195	119
PHARLAP 386|ASM/LINK	495	419
PHARLAP 386 DEBUG	195	155
SCO XENIX SYS V 386 (COMPLETE)	1495	1195
VM/386 *SPECIAL*	195	119
X-AM	595	535

ARTIFICIAL INTELLIGENCE

	LIST	OURS
ARITY STANDARD PROLOG	95	79
MULISP-87 INTERPRETER	300	199
PC SCHEME	95	85
SMALLTALK/V *NEW* V. 2.0	99	85
TURBO PROLOG	100	65
TURBO PROLOG TOOLBOX	100	65

ASSEMBLERS/LINKERS

	LIST	OURS
ADVANTAGE DISASSEMBLER	295	269
ADVANTAGE LINK	395	359
ASMLIB	149	125
EZ_ASM	70	65
MS MACRO ASSEMBLER	150	95
PASM86	195	109
PLINK86PLUS	495	275
RELMS, UNIWARE X-ASMS	CALL	CALL
VISIBLE COMPUTER 80286	100	89

BASIC

	LIST	OURS
DB/LIB	139	119
FLASH-UP	89	79
MACH 2	75	59
MS QUICKBASIC	99	65
QUICKPAK	69	59
TRUE BASIC	100	69
TURBO BASIC	100	65
DATABASE TOOLBOX	100	65
EDITOR TOOLBOX	100	65
TELECOM TOOLBOX	100	65

XENIX/UNIX PRODUCTS

	LIST	OURS
MICROPORT & SCO PRODUCTS	CALL	CALL
ADVANTAGE C++	695	CALL
BTRIEVE	595	455
C-TERP	498	379
INFORMIX ESQL/C	749	CALL
INFORMIX 4GL	1500	CALL
INFORMIX SQL	995	CALL
KORN SHELL	125	115
MICROSOFT LANGUAGES	CALL	CALL
PANEL	625	535
PANEL PLUS	795	675
REAL TOOLS	149	89
RM/COBOL	1250	949
RM/FORTRAN	750	549
SCO MULTIVIEW (286)	395	319
SCO MULTIVIEW (386)	495	399

C++

	LIST	OURS
ADVANTAGE C++	495	479
PFORCE++	395	209

C COMPILERS

	LIST	OURS
C86PLUS	497	375
LATTICE C	500	265
MICROSOFT C	450	269
QUICK C *SPECIAL*	99	59
TURBO C	100	65

C INTERPRETERS

	LIST	OURS
C-TERP	298	219
INSTANT C	495	369
RUN/C	120	79
RUN/C PROFESSIONAL	250	155

C LIBRARIES

	LIST	OURS
BASIC_C	175	129
C ASYNCH MANAGER	175	135
C-FOOD SMORGASBORD	150	95
W/SOURCE CODE	300	179
C TOOLS PLUS/5.0	129	99
C UTILITY LIBRARY	185	119
C-XPERT	395	339
ESSENTIAL COMMUNICATIONS	185	119
COMMUNICATIONS PLUS	250	189
GREENLEAF C SAMPLER	95	69
GREENLEAF COMM LIBRARY	185	125
GREENLEAF FUNCTIONS	185	125
MULTI-C	149	135
PFORCE *SPECIAL*	295	199
RESIDENT C W/SOURCE	198	169
TIMESLICER	295	265
W/SOURCE CODE	1000	895
TURBO C TOOLS	129	99

COBOL

	LIST	OURS
E-Z PAGE	295	259
MICRO FOCUS		
COBOL/2 *SPECIAL*	900	729
COBOL/2 TOOLSET *NEW*	900	729
PC-CICS	1500	CALL
LEVEL II COBOL	349	279
PERSONAL COBOL	149	119
OTHERS	CALL	CALL
MICROSOFT COBOL	700	429
MICROSOFT SORT	195	129
OPT-TECH SORT	149	99
REALCICS	995	785
REALIA COBOL	995	785
W/REALMENU	1145	899
RM/COBOL	950	759
RM/COBOL-85	1250	999
RM/SCREENS	395	315
SCREENIO	400	379

DEBUGGERS

	LIST	OURS
ADVANCED TRACE-86	175	115
C-SPRITE	175	119
PERISCOPE I	345	275
PERISCOPE II	175	139
PERISCOPE III 8 MHZ	995	799
PERISCOPE III 10 MHZ	1095	875
PFIX 86 PLUS	395	209
T-DEBUG PLUS	60	49
XVIEW86	60	49

DISK/DOS/KEYBOARD UTILITIES

	LIST	OURS
COMMAND PLUS	80	69
DISK OPTIMIZER	60	55
FETCH *SPECIAL*	55	45
INTELLIGENT BACKUP	150	135
NORTON COMMANDER	75	55
ADVANCED NORTON UTILITIES	150	99
PDISK	145	99
VFEATURE	80	75
VFEATURE DELUXE	120	110

EDITORS

	LIST	OURS
BRIEF	195	CALL
W/DBRIEF	275	CALL
EDIX	195	155
EMACS	295	265
EPSILON	195	149
KEDIT	125	99
PC/EDT	250	229
PC/VI	149	109
PI EDITOR *SPECIAL, NEW*	195	149
PMATE	195	109
SPF/PC	195	145
VEDIT PLUS	185	129

FILE MANAGEMENT

	LIST	OURS
BTRIEVE	245	185
XTRIEVE	245	185
REPORT OPTION	145	99
BTRIEVE/N	595	455
XTRIEVE/N	595	455
REPORT OPTION/N	345	269
CBTREE	159	139
C-TREE	395	315
R-TREE	295	239
C-TREE/R-TREE BUNDLE	650	519
DBC III	250	169
DBC III PLUS	750	595
DB_VISTA OR DB_QUERY	195	159
SINGLE USER W/SOURCE CODE	495	399
MULTIUSER	495	399
MULTIUSER W/SOURCE CODE	990	789
INFORMIX PRODUCTS	CALL	CALL
PHACT MANAGER	249	219
XQL *NEW*	795	599

FORTRAN COMPILERS

	LIST	OURS
LAHEY FORTRAN	477	CALL
LAHEY PERSONAL FORTRAN 77	95	89
MICROSOFT FORTRAN	450	269
RM/FORTRAN	595	479

FORTRAN UTILITIES/LIBRARIES

	LIST	OURS
DIAGRAM'ER OR DOCUMENT'ER	129	115
DIFF-E-Q	495	445
FORTRAN ADDENDA	165	139
GRAFMATIC OR PLOTMATIC	135	119
MAGUS NUMERICAL ANALYST	295	249
MATHPAC	495	445
NO LIMIT	129	109
SPINDRIFT LIBRARY	149	135
SSP/PC	350	269

GRAPHICS

	LIST	OURS
ADVANTAGE GRAPHICS (C)	250	225
ESSENTIAL GRAPHICS	250	185
GSS GRAPHIC DEV. TOOLKIT	495	375
HALO *SPECIAL*	300	199
HALO (5 MICROSOFT LANG.)	595	389
METAWINDOW PLUS	275	229
TURBOWINDOW/C	95	79
TURBO HALO (FOR TURBO C)	99	79

LINT

	LIST	OURS
PC-LINT	139	99
PRE-C	295	155

MODULA-2

	LIST	OURS
LOGITECH MODULA-2		
COMPILER PACK	99	79
DEVELOPMENT SYSTEM	249	199
TOOLKIT	169	139
WINDOW PACKAGE	49	39
ROM PACKAGE AND CROSS RUNTIME DEBUGGER	299	239
REPERTOIRE	89	75

TURBO PASCAL ADD-ONS

	LIST	OURS
DOS/BIOS & MOUSE TOOLS	75	69
FLASH-UP	89	79
METRABYTE DATA ACQ. TOOLS	100	89
SCREEN SCULPTOR	125	89
SYSTEM BUILDER	150	129
IMPEX	100	89
REPORT BUILDER	130	115
T-DEBUG PLUS	60	49
TURBO.ASM	99	69
TURBO ASYNCH PLUS	129	99
TURBO EXTENDER	85	65
TURBO HALO	99	79
TURBO MAGIC	199	179
TURBO OPTIMIZER	75	65
TURBO POWER TOOLS PLUS	129	99
TURBO POWER UTILITIES	95	79
TURBO PROFESSIONAL 4.0 *NEW*	99	79
TURBO WINDOW/PASCAL	95	79

FEATURED PRODUCTS

MICROSOFT WINDOWS/386—Software for 386-based machines offering true multi-tasking. Run several programs within different windows on your screen. Cut and paste between them. Each program is given 640K of memory.
List: $195 **Special Price: $119**

MICRO FOCUS COBOL/2—Compiler system that can exploit the full memory of 80286/386 machines under OS/2 or under PC-DOS. Conforms to the highest certifiable level of ANSI'85 COBOL. Includes the ANIMATOR source code debugger and offers full network support.
List: $900 **Special Price: $729**

PANEL/QC or TC—This best selling screen management library is now available for both Quick C and Turbo C. Supports pop-up fields and windows, multi-line fields, horizontal and vertical field scrolling, menus, help boxes, and custom field validation. Generates C source code. No royalties.
List: $129 **Special Price: $89**

FETCH—New memory resident file librarian, for those of us who have trouble finding files, or forget what's in them. The user is prompted for a 255 character description anytime a file is created. Later, Fetch can scan the file description library with its fast pattern recognition capabilities, and will display the description, directory, and drive.
List: $55 **Special Price: $45**

OPERATING SYSTEMS

	LIST	OURS
MICROPORT SYSTEM V/AT	549	465
SCO XENIX SYSTEM V	1295	989
WENDIN-DOS	99	79
OTHER MICROPORT, SCO, WENDIN PRODUCTS	CALL	CALL

PASCAL COMPILERS

	LIST	OURS
MARSHAL PASCAL	189	155
MICROSOFT PASCAL	300	185
PASCAL-2	350	319
TURBO PASCAL *NEW* V. 4.0	100	65
TURBO PASCAL DEV. LIB. *NEW*	395	259
BORLAND ADD-ONS	CALL	CALL

SCREEN DISPLAY/WINDOWS

	LIST	OURS
C-SCAPE	279	265
CURSES W/SOURCE CODE	250	169
GREENLEAF DATA WINDOWS	225	155
W/SOURCE CODE	395	259
HI-SCREEN XL	149	119
JYACC FORMAKER	495	449
JYACC JAM	750	679
MICROSOFT WINDOWS	99	65
MS WINDOWS DEVELOPMENT KIT	500	309
PANEL PLUS	495	395
PANEL/QC OR TC *SPECIAL*	129	89
QUICKSCREEN	195	175
SCREENSTAR W/SOURCE	198	169
VITAMIN C	225	149
VC SCREEN	99	79
VIEW MANAGER	275	199
WINDOWS FOR DATA *SPECIAL*	295	229

HARDWARE PRODUCTS

NEW

	LIST	OURS
AMDEK 722 MONITOR	750	499
AMDEK 730 MONITOR	899	569
AST ADVANTAGE PREMIUM W/512K	495	319
AST RAMPAGE! 286 W/512K	545	349
HERCULES GRAPHICS CARD PLUS	299	195
HERCULES IN COLOR CARD	499	329
IRMA 2	1195	779
ORCHID TURBO EGA	749	495
ORCHID TURBO PGA	1495	1099
PARADISE SYSTEMS		
VGA PROFESSIONAL	599	409
AUTOSWITCH EGA 480 CARD	349	169
VEGA DELUXE	379	259

ADDITIONAL PRODUCTS

	LIST	OURS
ADVANTAGE VCMS	379	329
BASTOC	495	399
CARBON COPY PLUS	195	159
DAN BRICKLIN'S DEMO PROGRAM	75	59
DB2C	299	CALL
FLOW CHARTING II	229	205
MAGIC PC	195	179
MKS TOOLKIT	139	115
NORTON GUIDES	100	65
PFINISH	395	209
POLYMAKE	149	125
POLYTRON PVCS	CALL	CALL
SOURCE PRINT	95	75
TREE DIAGRAMMER	77	69

Terms and Policies
- **We honor MC, VISA, AMERICAN EXPRESS** No surcharge on credit card or C.O.D. Prepayment by check. New York State residents add applicable sales tax. Shipping and handling $3.00 per item, sent UPS ground. Rush service available, prevailing rates.
- Programmer's Paradise will match any current nationally advertised price for the products listed in this ad.
- Prices and Policies subject to change without notice.
- Hours 9AM EST—7PM EST

*Ask for details. Some manufacturers will not allow returns once disk seals are broken.

Corporate Buyers—Call for special discounts and benefits!

1-800-445-7899
In NY: 914-332-4548
Customer Service: 914-332-0869
International Orders: 914-332-4548
Telex: 510-601-7602

CIRCLE NO. 173 ON READER SERVICE CARD

Programmer's Paradise™
A Division of Hudson Technologies, Inc.
42 River Street, Tarrytown, NY 10591

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PRODUCT CATEGORIES

Advertising Rates and Information:
PC Tech Journal Marketplace

PC Tech Journal Marketplace is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches. Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $220.00 per issue ($660 total).

For additional information call 212-503-5115.

PC Tech Journal Classified Advertising Staff
One Park Avenue, New York, NY 10016
(212) 503-5115

Advertising Director
Kathryn J. Cumberlander

Sales Manager
Kathleen F. Lyman

Sr. Advertising Coordinator
Monica Dixon

Advertising Coordinator
Ana Marie Gonzales

Production Manager
Anne R. Brockinton

Stanley H. Robinson, Account Manager
(212) 503-5116
ME, CANADA – (OTHER THAN BRITISH COL.), NY, NJ, PA, MA, CT, VT, NH, RI, WV, MD, VA, NC, SC, GA, FL, DC, DE

Ariane R. Casey, Account Representative
(212) 503-5172
WA, HI, AZ, MT, NV, ID, ALL CALIF, NM, UT, CO, OR, WY

Lee J. Uniacke, Account Representative
(212) 503-5141
ND, KS, NB, OH, OK, IA, SD, KY, MI, AL, IL, LA, MN, WI, TN, MS, TX, AR, BRITISH COL., MO, ALL OVERSEAS CALLS.

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

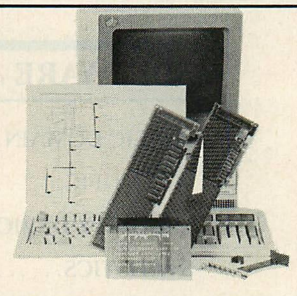
Hardware/Accessory Cards

TMS 32020 PC COPROCESSOR BOARD

- 20 MHz, 5 MIPS, Addressing 256K
- 1 Clock 16 Bit Multiply
- Ideal for High Speed Numerical Analysis, Graphics, and DSP
- External User BUS for A/D & D/A
- Monitor/Debugger & C Utilities
- Fractals, Rotations, FFT Demos and Source Included

$900 board & 32K
$1100 board & 256K
$150 TMS32020 assembler

SYMMETRIC RESEARCH
15 Central Way, Suite 9, Kirkland, WA 98033
206-828-6560
CIRCLE 265 ON READER SERVICE CARD



PS/2™ Compatible Products
Everything for PS/2 machines

- WIREWRAP CARDS high quality design in 6 sizes
- EXTENDER CARDS
- BRACKETS
- DESIGN EXAMPLES
- TECHNICAL DETAILS
- ASIC CHIPS
- PS/2 NEWSLETTER

- CALL FOR OUR FREE CATALOG

ONTARGET
TARGET
TARGET
TARGET

ON TARGET Associates
1250 Oakmead Parkway #210
Sunnyvale, CA, 94086-3599
(408) 980-7118
PS/2 is a registered trademark of IBM Corp.
CIRCLE 267 ON READER SERVICE CARD

Truly Low Cost PC Imaging!

IMAGE ACE II
Video Capture System
- Digitize video from cameras, tuners, and VCRs directly to your IBM PC display
- 320 x 200 x 4 levels
- 1.3 sec. full screen capture
- Complete with hardware card, software, cable, and manual

LODGE ELECTRONICS
P.O. Box 338 • Streamwood, IL 60103
(312) 837-6553
CIRCLE 266 ON READER SERVICE CARD

PC TECH JOURNAL MARKETPLACE is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches.

Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $220 per issue ($660 total).

DIGITAL SIGNAL PROCESSOR
The Model 10 coprocessor board is based on the 16/32 bit TI TMS 32010 and is designed for applications in communications, speech, instrumentation, and numeric processing. A 1K complex FFT takes 90ms. Offered with onboard 12 bit 80 Khz A/D and D/A. Continuous data acquisition & playback option. Includes all utility and applications software. $650 & up.
DALANCO SPRY
Suite 241 2900 Connecticut Ave. NW
Washington, DC 20008
(202) 232-7999

8MB FOR YOUR AT
* uses 1M X 1 DRAMs
* 1 serial & 1 parallel port
* configured through software
* can be used as base or extended memory
0MB - $495; 2, 4, 6, 8MB CALL
ADVANCED CIRCUIT DESIGN
3932A 93rd Avenue SW
Olympia, WA 98502
(206) 352-4810

TECH MARKETPLACE . . .
The Comprehensive Guide to Products and Services for the MS DOS Market
(212) 503-5115

Communications

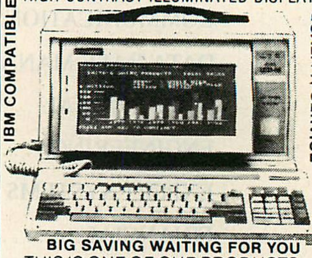
RS-422
COMMUNICATIONS BOARD FOR
PS/2
MODELS 50, 60, 80
* DUAL CHANNEL
* ADDRESS SELECTABLE
* BAUD RATE TO 256 K
* INTERRUPT SELECTABLE

1-800-553-1170

QUA TECH INCORPORATED
Leaders in Communication Technology
478 E. EXCHANGE ST.
AKRON, OH 44304
(216) 434-3154 TLX: 5101012726
FAX: (216) 434-1409
CIRCLE 268 ON READER SERVICE CARD

Compatibles

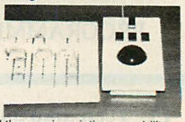
Portable AT/386
HIGH CONTRAST ILLUMINATED DISPLAY
IBM COMPATIBLE
QUALITY, SERVICE



BIG SAVING WAITING FOR YOU
THIS IS ONE OF OUR PRODUCTS
ASK FOR LATEST CATALOGUE
MAXTRON (818) 350-5707
1825A Durfee Ave., S. El Monte, CA 91733
CIRCLE 269 ON READER SERVICE CARD

Peripherals

FastTRAP™
The pointing device of the future is here!



- Two and three axis pointing capability
- High resolution trackball for X and Y axis input
- High resolution fingerwheel for Z axis input
- Use with IBM® PC's, XT's, AT's and compatibles
- Three input buttons
- Full hardware emulation of Microsoft® Mouse
- Standard RS-232 serial interface
- Includes graphics drivers and menu generator
- Easy installation
- 1 year warranty
- Made in the U.S.A.

ONLY **$149.00**
VISA and MasterCard accepted

LTS/C Corp.
319 South Limestone Street
Lexington, Kentucky 40508
(606) 233-4156
(800) 872-7279

3-D TRACKBALL FOR IBM
CIRCLE 270 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PERIPHERALS

9-Track Tape Subsystem

MINISTREAMER™

Now you can exchange data files between your IBM PC/XT/AT and any mainframe or mini-computer using IBM compatible 1600 BPI 9-Track tape. Unit can also be used for disk backup. Transfer rate is up to 4 megabytes per minute on PCs and compatibles. Subsystems include 7" or 10 ½" streaming tape drive, tape coupler card and DOS or XENIX compatible software.

Prices start at $2,995.

QUALSTAR®

9621 Irondale Avenue
Chatsworth, CA 91311
Telephone: (818) 882-5822

CIRCLE 271 ON READER SERVICE CARD

SPEECH SYNTHESIS

SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™

Electronic Speech Articulator

Artic Technologies
55 Park St., Suite 2
Troy, MI 48083
(313) 588-7370

SOFTWARE

BUSINESS

ACCOUNTING FOR YOUR VERTICAL SOFTWARE

Complete, integrated accounting system, in C (using C-Tree file manager) including General Ledger, Accounts Receivable, Accounts Payable and Payroll as a base for vertical package or standalone. No monthly or yearly closing required. Extremely flexible for operator, 5 years in development, 50,000 lines of code.

First Resort Software, Inc.
715 W. Main
Aspen, CO 81611
(303) 925-5481

BUSINESS

LP88—SPREADSHEET LP

Our best-selling linear programming system solves problems w/1000 constraints and 5000 variables, reads/writes Lotus worksheets, many other advanced features. IE News says "the flexibility and features of this program are a bargain at its low price." Req. 192k. $149 w/manual and 8087 support. $29 demo.

Eastern Software Products, Inc.
P.O. Box 15328
Alexandria, VA 22309
(703) 360-7600

COMMUNICATIONS

ARC - MAP

Asynch RS-232 Comm - Monitor/Analyzer Prog Use PC to diagnose data communications. With cables, manual; 2-19200 BPS; 32K bfr; data, errors, mdm sig's; Passive, Active, BERT; ASCII, EBCDIC, Hex, Dec, Oct; bfr search; event traps; timer; save strings, setups, data; EASY use!! $249, 30 day M/B

George Driver
1316 El Rancho Dr.
Bakersfield, CA 93304
(805) 835-1916

DOS COM PORT DEVICE DRIVER

Interrupt driven device driver used for the COM ports on PC, AT & compatibles. With source code in Assembly & test program in C. Use instructions as open, close, read, write & ioctl to access your COM port data. Load driver through config.sys at boot up. $39.95 with manual.

HYTEC RESEARCH, INC.
20705 S. Western Ave., Suite 221
Torrance, CA 90501
(213) 320-4541

DATA BASE MANAGEMENT

QuickBASIC DOES DATABASE!

db/LIB™

New! Enhance QuickBASIC with Relational Database Management!

You get 3 Application-Building Libraries that:

- ☑ Read and write dBASE III + files.
- ☑ Build indexes for any files.
- ☑ Execute data expression macros.
- ☑ Include dozens of high-level routines.

Create your application programs with just block move and edit!

Write for brochure or send $3.00 for demo disk (refundable).

Only ***$139*** complete.

AJS Publishing, Inc.

P.O. Box 379 North Hollywood, CA 91603

Available at leading software stores.

CIRCLE 272 ON READER SERVICE CARD

DATA BASE MANAGEMENT

FREE

dBASE III + COMPILER DEMO

DON'T BUY CLIPPER™ UNTIL...

you see our free dBASE Compiler Evaluation Kit. Contains a demo disk with 8 PRG's and the results of compiling with Clipper, Quick-Silver, and FoxBASE PLUS. See for yourself how all three compilers work. Also 15 benchmark tests, detailed brochures, and complete reprints of magazine reviews about all three. FREE. NO OBLIGATION. Call 24 hrs.

dataBASE Specialties
P.O. Box 2975
Oakland, CA 94618
(415) 652-2790

CIRCLE 273 ON READER SERVICE CARD

GRAPHICS

FORTRAN GRAPHICS LIBRARY

GRAFMATIC (screen graphics): 75 MS FORTRAN/Pascal, R-M/Profort, Lahey FORTRAN callable subroutines. Fully documented, prof. graphics capabilities, inc. general utility, 2-D interactive, total 2-D plots, 3-D plots and solid models. $135. H-P or H-I plotter? get PLOTMATIC, complete plotter graphics library. Interfaces w/GRAFMATIC. $135. Both $240.

MICROCOMPATIBLES, INC.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Use high resolution color slides up to 4000 line. COMPUTER SLIDE EXPRESS $9/slide.

VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

Tech Marketplace...

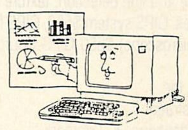
The comprehensive guide to products and services for the MS DOS market.

To place your ad Call

(212) 503-5115

GRAPHICS

PlotIT®



Interactive Graphics & Statistics

Create graphs the way you want them. 35 graph types can be used to create numerous renditions. Make posters and flow charts—with full screen editing of text.

- **Vary character font, size, position, color**
- **Stack and overlay graphs**
- **3-Dimensional graphs: fishnet & contour**
- **Linear & Non-Linear Regression**
- **Stochastic Distribution**
- **Independence Testing**
- **Descriptive Statistics**

Contact: Scientific Programming Enterprises, P.O. Box 669, Haslett, MI 48840 (517) 339-9859

CIRCLE 274 ON READER SERVICE CARD

QUICK SCIENTIFIC GRAPHICS

QuikPlot is a fast, powerful, program for producing serious scientific and engineering plots on your screen from data files. No programming required.

- Multi-curve, scatter & 3-D plotting
- Plots up to 3600 points in 60 curves
- Plots up to a 60 X 60 grid in 3-D
- Automatic or manual axis scaling
- Save plots to disk and retrieve
- Well-documented and easy to use

Hard copy obtained through screen dump. For CGA or equiv. Not copy protected. $29.95 + 2.50 shpg.

THE BALLENTINE COMPANY
P.O. Box 2036
Culver City, CA 90231-2036

PLOTTER EMULATION (w/VGA)

FORTRAN callable, Versatec/Calcomp compatible plot subroutines for VGA, EGA, CGA, Herc. video and Epson and compatible printers at optimum resolution. Contour and 3d plots. 13 character fonts. Libraries for MS 3.3/4.0 and Lahey F77L, 150 page manual with examples. Only $200, HP plotter driver $50. Educational discounts.

F and S Software
7604 Peacock Drive
Huntsville, Alabama 35802
(205) 881-6268

GRAPHICS

Category

continued on next page

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

GRAPHICS

GIPS Image Processing
Comes complete with manual and tutorials. Runs on PCVision Plus, Number Nine and AT&T VISTA boards. Features include basic operations, geometry, edge and line detection, texture analysis etc. Prices: GIPS system $550, GIPS feature Analysis (microscope) $550 GIPS Sourcecode $5000.
PC-Expand, Inc.
St. Kannikestraede 7
DK 1169 Copenhagen K, Denmark
Tel. +45 1 156800, FAX +45 1 939901

LANGUAGES

FIRMWARE DEVELOPMENT IN C
REX-C/86 C package supports ROM code generation. Includes XC86 C compiler which implements draft ANSCI C standard, supports-in-line assembly, produces optimized 8086/87/186 reentrant code for real-time environment, generates separate segment for initialized data and string constants for ROM-based applications, produces object file in Intel OMF with debug information, global and local symbols with data type and line numbers for high-level debugging. Price is $750 which includes XC86, linker, locator, librarian, hex formatters, and run-time library source.

SYSTEMS & SOFTWARE! FIRMWARE DEVELOPMENT

SYSTEMS & SOFTWARE, INC.
3303 Harbor Blvd., C11
Costa Mesa, CA 92626
(714) 241-8650

FORTRAN 77 EXTENSION
EXTEND™ routines for MS, RM, IBM Pro compilers to control keyboard, monitor, file & directory operations, parallel & serial I/O user graphics for CGA, EGA, VGA, HP7475A, TEK4010, AutoCAD DXF & db files. Both 8087 & non 8087 libs pkgd for $149. No royalty. MC/VISA/Chk/MO.
DESIGN DECISIONS, INC.
P.O. Box 12884
Pittsburgh, PA 15241
(412) 941-4525

NETWORKING

KEYCARD ELIMINATOR

NOVELL ADVANCED NETWARE®
V2.0a
86, 286, 286 nonded, SFT-1, SFT-2

SLOTBOUND?
FREE A SLOT IN YOUR FILESERVER!
$99.00 (includes S & H)
M/C, Visa or C.O.D.
TX Residents add 7.25% Tax
CALL FOR DEALER PRICING

NBS
NETWORK BUSINESS SYSTEMS
1300 Woodhollow Drive, Suite 5601
Houston, Texas 77057
713-781-9268 (Sales, Tech Support)
713-783-4457 (Administrative, Sales)
* NetWare is a Trademark of Novell, Inc.
CIRCLE 276 ON READER SERVICE CARD

NETWORK CONTROL LIBRARIES
NETWORK INTERFACE allows file sharing and redirection through DOS functions. $99. NETBIOS ROUTINES allows access to low-level network functions. Name, session & datagram routines. Wait and no-wait options. $199. NETWORK MASTER provides access to Netware internal functions. Complete control of your network from your compiled programs.
Starlight Software
2861 Central Street
Evanston, IL 60201
(312) 864-9370

NOVELL NETWORKING PRODUCTS
When reliability and performance really count you can't afford anything less than Novell. We offer quality Novell products and accessories, fast courteous service and competitive pricing for all your networking needs. Our experienced staff is waiting to serve you. Call today for information and quotes.
BENEDEK ENTERPRISES
104 Cool Springs Road
White Oak, PA 15131
412-751-8381

Tech Marketplace
Standard Directory
Listings Available
Call (212) 503-5115

LANGUAGES

** Introductory Special **
FTL MODULA-2 — ONLY $79.95

Large Memory Model Modula-2 Compiler for MSDOS Systems.
Includes: Compiler, Linker, Editor, 8087 support, and full Library sources.
FTL Modula-2 can create programs as large as your available memory.

Don't Delay Order TODAY!!

SAVE $10.00 when you order one or more of the support packages.
Editor Toolkit $39.95
Small Compiler $49.95
8087 support (SMM) $29.95

Workman & Associates
1925 E. Mountain Street
Pasadena, CA 91104
(818) 791-7979

CIRCLE 275 ON READER SERVICE CARD

OPERATING SYSTEMS

AMX 86

KADAK's engineers bring years of practical real-time experience to this mature

MULTI-TASKING SYSTEM
(version 2.0)
for the IBM® PC, PC/XT and PC/AT

- No royalties
- IBM PC DOS® support
- C language support
- Preemptive scheduler
- Time slicing available
- Intertask message passing
- Dynamic operations:
  - task create/delete
  - task priorities
  - memory allocation
- Event Manager
- Semaphore Manager
- Source code of the C interface and device drivers is included

Demo package **$25 US** (shipping/handling extra)
Manual only **$75 US**
AMX86 system **$2195 US**
Also available for 8080, Z80, 68000

AMX86™ operates on any 8086/88, 80186/88, 80286 system.

KADAK Products Ltd.
206-1847 W. Broadway, Vancouver, B.C., Canada V6J 1Y5
Tel: (604) 734-2796 Telex: 04-55670

CIRCLE 277 ON READER SERVICE CARD

PROGRAMMERS TOOLS

SOURCER™

Create detailed commented source code and listings from memory, .COM files or .EXE files directly suitable for assembly. Built in data analyzer and simulator resolves multiple data segments and provides detailed comments on interrupts and subfunctions, I/O ports and much more.

BIOS SOURCE

PS/2 ■ AT ■ XT ■ PC ■ Clones

The bios pre-processor to SOURCER™ provides the first means to obtain accurate legal source listings for any bios! Identifies entry points with full explanations. Provides highly descriptive data labels such as "video_mode" and much more.

SOURCER	$ 99.95
SOURCER w/BIOS Pre-Processor	$139.95

To order or receive information just call!
800-538-8157 x 811 (outside Calif.) **800-672-3470 x 811** (inside Calif.)

V COMMUNICATIONS
3031 Tisch Way, Suite 200, Dept. TJ
San Jose, CA 95128 • (408) 296-4224

PS/2, AT, XT, PC are trademarks of IBM Corp.
CIRCLE 278 ON READER SERVICE CARD

TECH MARKETPLACE
HOME OF THE
POWER BUYER

PROGRAMMERS TOOLS

4 Times Faster than MASM 5.0

All MASM features (except 386 & CodeView support)
Plus,
*Generates smaller code! (no inserted NOP's)
*Automatically handles jumps out of range
*Handles most of MASM's phase errors
*Built in MAKE
*Up to 15,000 symbols
*Simplified segmentation

OPTASM $195

SLR Systems
1622 N. Main St. Butler, PA 16001
(412) 282-0864 (800) 833-3061
TELEX 559215

CIRCLE 279 ON READER SERVICE CARD

C & MASM TOOLS

VersiMAKE™
A full-featured MAKE utility that derives your system's dependencies through analysis of your C & MASM source files. No more MAKE dependency files to maintain!

VersiCREF™
A unique utility that creates a sorted Master Cross-Reference of your entire system. Handles 100+ C and MASM source files. Full X-Ref or just PUBLIC symbols.

VersiMAKE™	$125
VersiCREF™	$75
Both	$150

800-334-4096
(In NJ, 609-871-0202)
MC/VISA/AMEX

SUMMIT INFORMATION SYSTEMS, INC.
73 East Lane, Willingboro, NJ 08046
CIRCLE 280 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

TURBO PACKAGE 3.0 (for Turbo Pascal)

MULTI-TASKING, MULTI-PROCESSING, & MODULAR REENTRANT 640K PROGRAMS

FREE: A multi-processor data entry program for 2 or more machines.

Don't pay high prices for a network. Turbo Package has true multi-processing. Use all the power and memory you need.

A file processor LAN only gives you shared disk drives and a print spooler. Turbo Package lets you run data entry on several PC's, number crunching on a 386, intelligent file processing on an AT/XT, and print spooling. Hardware required: one PC or compatible. Any MIMDA configuration is possible.

Balance your work load and reduce hardware costs. Network with a modem (Wide Area Network), RS232, or RS422 (Local Area Network). Has NET BIOS support.

Few code changes required. Call 640K programs, single processor tasks, and multi-processor tasks just like procedures and functions.

You also get ZapIO, a high speed screen IO that works with writeln().

Turbo Package 2.0
w/ 640k Pascal programs $59.95
w/ Multi-Tasking 89.95
Turbo Package 3.0
w/ Multi-Processing 149.95
SuperMath 32-bit toolbox
40+ math routines 24.95

90 day money back guarantee (In Texas add TAX)
Visa, M/C, or COD Shipping: USA $5
Foreign $10

Conversational Computer Systems
5371 Verbena Rd.
San Antonio, TX 78240
Ph: (512) 692-0353

CIRCLE 281 ON READER SERVICE CARD

New ISAM for C

- FAST B-Tree structure.
- Open, Lookup, Insert, Delete, Next Key, Previous Key.
- Multiple indexes per data file.
- Multiple keys per record.
- Duplicate keys optional.
- Fast **CINDEX** utility.
- Variable-length data records.
- Documentation & sample program.

NEW . . . only from:
Infopoint Systems, Inc.
P.O. Box 851572 • Richardson, TX 75085
(214) 669-4700
VISA/MC Accepted $129 + $6 S & H
CIRCLE 286 ON READER SERVICE CARD

Turbo GhostWriter

TPascal Generator
Starter Kit
Now Available **$99**

B-tree File Mgr-No 64K limit
Screen Editor/Painter
Create a Relational
100% Complete Database

MC/Visa/COD $289
Normal Price $450
ASCII 800-227-7681
3239 Mill Run, Raleigh, NC
CIRCLE 287 ON READER SERVICE CARD

PLOT TEXT ON ANY GRAPHICS SCREEN!!

YES, we said ANY Graphics Screen, even VGA! FINALLY! Xgraf is a super set of smart low level assembly graphic routines that you call directly from Compiled BASIC. Xgraf replaces BASIC's confusing graphics statements with consistent, full featured calls specifically designed for the BASIC programmer.

FINALLY! Xgraf is only $99.00 + $4.00 S&H

We specialize in libraries and tools for Compiled BASIC. Our catalog features the FINALLY! Family of Products and other top flight tools.

Call: 1 800 423-3400
(9:00 AM to 8:00 PM EST)
PA & AK call (412) 782-0384

KOMPUTERWERK
851 Parkview Blvd.
Pittsburgh, PA 15215

CIRCLE 282 ON READER SERVICE CARD

MULTI-TASKING TOOLKIT

Write your 'C' programs with cooperating, concurrent tasks using INTERWORK. Very useful for parallel programming, simulation, and real-time applications. INTERWORK supports more than 100 tasks, provides inter-task communication facilities, and handles DOS interrupts. PC/DOS version $129, Xenix $159, Unix $249.
Block Island Technologies
13563 NW Cornell Road, Suite 230
Portland, Oregon 97229-5892
(503) 241-8971

SCREEN MANAGER

MENU, WINDOW, and DATA ENTRY Support for the Professional Programmer! Interfaces to most languages. BASIC, C, FORTRAN, COBOL, PASCAL, ASSEMBLER. 100 Page Manual. Thirty day money back guarantee. No Royalties.

from **$79** VISA/MC

The West Chester Group
P.O. Box 1304
West Chester, Pa 19380
(215) 644-4206

CALL FOR FREE DEMO
CIRCLE 288 ON READER SERVICE CARD

One-stop ROM Shop

When your application calls for ROM support, contact the professionals at ALDIA SYSTEMS. We've been supporting firmware developers for over two years with excellent software development tools and technical support. Don't settle for imitations, call the company who brought you ***PC-LOCATE***, the original PC locater.

PC-LOCATE: Produce ROM-able code from your ".EXE" files. PC-LOCATE assigns physical addresses to the re-locatable image based on user inputs. PC-LOCATE support the entire Intel processor family including : 8086, 8088, 80186, 80188 and 80286.

PC-PROMPAK: A PROM/ROM expansion board for IBM and IBM-compatible computers. PC-PROMPAK provides up to 384Kbytes of non-volatile expansion memory and supports most 28-pin JEDEC devices including EPROMS, EEPROMS and Static RAMs.

PC-ROMDRIVE: Create a "Diskless PC" that can include MS-DOS and your application program. "Autoexec.bat" files are supported for automatic program execution.

ALDIA SYSTEMS, Inc.
P.O.Box 37634 Phoenix, Az. 85069
(602) 866-1786
CIRCLE 283 ON READER SERVICE CARD

BASIC BASE

For Business Applications

Basic Base has the speed and power of 4GL in the BASIC you know and love. Basic Base is used by both people and programmers in database applications. People use the utilities for rapid data entry and reporting. Programmers use the library, which contains all the necessary screen and data access routines, for developing applications with:

Menus	4 Indexes per File
Passwords	Edited data entry
Data Schema	Function key usage

Past Developments Include:

Inventory	Purchasing
Vouchers	Personnel

$199
—30 day money back guarantee.—

No Royalties. No copy protection.
Application Micro Computers Inc.
1663 Bachan Ct.
Reston, Va. 22090
703-471-1471
CIRCLE 284 ON READER SERVICE CARD

CMS UPDATE FOR THE PC

UPDATE/PC is a source maintenance program modeled on VM/CMS UPDATE. Also included is an update-generator which works with most PC editors to imitate XEDIT (UPDATE). Supports variable-length records, multiple auxiliary files, line# or sequence# modes. PC/XT/AT with 512K, DOS 2+.$75, CT + 7.5%.
Monopole Systems, Inc.
12 Cardinal Road
Weston, CT 06883
(203) 226-9229

Easy Assembly Maintenance

Debug, develop and maintain assembly language faster and easier with PAL structured concepts. IBM/MASM compatible. Use more powerful commands and new verbs, including: If, select, do include and leave. $50.
LANEY SYSTEMS, INC.
3 Office Park Drive, Ste. 100
Little Rock, AR 72211
501-225-7755

YOUR MANUAL—TYPESET

KN Systems will design, lay out, and typeset your user's manual or other documentation from your machine readable text file. Returned typeset, not Laser printed; ready for printing.
Choose from over 1000 fonts in sizes from 5 to 72 point. You specify page size, margins, everything! Advance proofs included.
KN Systems
6114 LaSalle Ave., Suite 255
Oakland, CA 94611
(415) 339-3606

COMPLEX FILE I/O PROBLEMS?

Heap I/O is a library which enables you to treat all or part of a disk file as a heap memory. You may allocate, free, read and write variable length blocks of storage in a file. Your data structures on disk can be as rich as those you create in memory. Microsoft C object $59.95. For source add $79.95.
Roland Alden System Software
One Pine Street, Suite 2509
San Francisco, CA 94111
(415) 397-9316

ROM YOUR ".EXE"

Produce a file suitable for loading into a ROM from MS-DOS ".EXE" file. Generates binary file located to user requested segment. Includes utility to create Intel Hex Format ascii file. Menu-driven or command-line. U.S. funds—$29.95 plus $3 s/h. CHK/MO
VIRTUAL SOFTWARE
51 OAK AVE.
RICHMOND HILL, ONTARIO, CANADA L4C 6R5
(416) 881-7665

CROSS ASSEMBLERS

Macros, PC Compatible, Linker, Editor, Relocatable, Conditionals, Fast, Reliable. For most microprocessors . . . from $150
also: Cross Debug/Simulators
EPROM Programmer Board
MICROCOMPUTER TOOLS CO.
912 Hastings Dr.
Concord, CA 94518
Phone (800) 443-0779 In CA (415) 825-4200

TECH MARKETPLACE . . .

Standard directory listings available. To place your ad

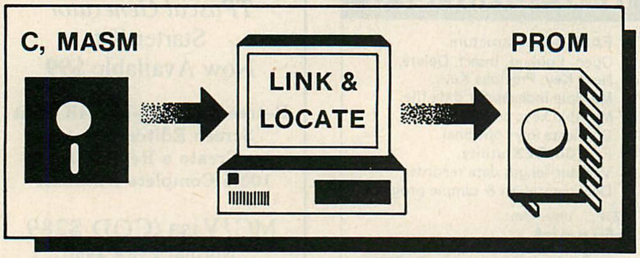
Call (212) 503-5115

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

Programmers Tools

FIRMWARE DEVELOPMENT



LINK & LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by popular C compilers, such as from Microsoft, Lattice and Borland's Turbo C, and MASM from Microsoft, Provides full control of segment placement anywhere in memory. Supports output of Intel HEX file for PROM programmers, Intel OMF absolute object file for symbolic debuggers and in-circuit emulators. Includes Intel compatible linker, locator, librarian, hex formatter and cross reference generator. $350.

NEW! Includes utility to support PC based PROM programmers.

SYSTEMS & SOFTWARE INC.

3303 Harbor Blvd., C11, Costa Mesa, CA 92626

Phone (714) 241-8650 FAX (714) 241-0377 TWX 910-695-0125

CIRCLE 289 ON READER SERVICE CARD

NEW! TLIB™ 4.0 VERSION CONTROL SYSTEM

The best keeps getting better!

The critics *loved* TLIB 3.0: "...packed with features... [generates deltas] amazingly fast... [of 6 reviewed] the two best packages are Burton Systems' TLIB and [a $395 product], so designated because of their ease of use, abundance of features, and ability to be configured..." ***PCTJ Sept 87***

"...has my highest recommendation." *Ronny Richardson,* ***Computer Shopper Aug 87***

- ***Many new features!*** Expanded keyword support. Multi-line comments. Branching, for multiple development lines. Extended wildcard and list-of-file support; creates lists by scanning source code for includes. Can merge (reconcile) multiple simultaneous changes and undo intermediate revisions. Network and WORM optical disk support. Now even faster!
- Includes Landon Dyer's excellent public domain **MAKE** (with source).

PC/MS-DOS 2.x & 3.x ***Just $99.95*** + *$3 s/h* Visa/MC

BURTON SYSTEMS SOFTWARE

POB 4156, Cary, NC 27519 **(919) 469-3068**

CIRCLE 290 ON READER SERVICE CARD

Scientific

SCI/ENG GRAPHICS

OMNIPLOT [S] (screen graphics) & OMNIPLOT [P] (plotter driver) provide integrated engineering/scientific 2-D & 3-D graphics with NO PROGRAMMING! Menu-driven, flexible, professional. Choice of formats: tabular/line, contour, bar, pie, 3-D wire frame & much more! OMNIPLOT [S] $195. Add OMNIPLOT [P], both $295.
MICROCOMPATIBLES, INC.
301 Prelude Dr. Dept. J
Silver Spring, MD 20901
(301) 593-0683

8087 FFT/VECTOR PROCESSING

The VECTOR87 library is written in assembler, includes 60 routines to speed up your number-crunching programs, uses 80(2)87 extensively. PC 1K real FFT takes only 1.2 sec. Versions for Fortran (MS, RM, Lahey), C (MS, Lattice), Turbo Pascal -87. $150 per version with source, no royalties. Write for technical information.
VECTORPLEX Data Systems Ltd.
136-100 Maitland Place N.E.
Calgary, Alberta, Canada T2A 5V5
(403) 248-1250

Security Devices

EVERLOCK COPY PROTECTION

Designed for user-transparency, clone compatibility & strength. It features:

- no need for damaged media or I/O plugs
- supports all Hard & Floppy disk formats
- file-server networks supported
- variable number of installs (0-99)
- create demos with remote unlock option
- allows protected upgrades by modem/BBS

$195 Starter Kit or $495 with NO meter counts. Free info & demo disk available.

Az-Tech Software, Inc.
305 East Franklin
Suite A4A
Richmond, MO 64085
(800) 227-0644 . . . (816) 776-2700

CIRCLE 291 ON READER SERVICE CARD

BIT-LOCK® SECURITY

Piracy SURVIVAL 5 YEARS proves effectiveness of powerful multilayered security. Rapid decryption algorithms. Reliable/small port transparent security device. PARALLEL or SERIAL port. Countdown and timeout options also available. KEY-LOK™ security at about ½ BIT-LOCK cost.
MICROCOMPUTER APPLICATIONS
7805 S. Windermere Circle
Littleton, CO 80120
(303) 922-6410 or 798-7683

Tech Marketplace
The home of the Power Buyer

Statistics

RATS! New Version 2.10

Full function regression, time series and forecasting program. Multiple regressions, including stepwise. Non-linear least squares. ARIMA models, transfer functions & intervention analysis. Exponential smoothing, spectral analysis, Kalman filter, model simulations and much more! High-quality graphics to screen, plotter, printer. $200-$300. Visa/MC.
VAR Econometrics, Inc.
P.O. Box 1818
Evanston, IL 60204-1818
(312) 864-8772; (800) 822-8038

Terminal Emulation

BARR High-Performance RJE Workstations

BARR/SNA RJE and BARR/HASP are communications hardware/software packages for the IBM PC and PS/2 series of computers.

Full featured:
–multiple printers,
–unattended operation,
–print speed beyond 6,000 lpm,
–special forms,
–line speed to 56,000 bps,
–dial-up or dedicated lines, and
–serve as a LAN gateway.

BARR/SNA RJE emulates IBM 3777-3 in an SNA environment. BARR/HASP emulates IBM 3777-2 and HASP on the 360/20. BARR communications software is also available with PC-SYNC internal modems: 208AB, 201C, 9600.

BARR

Barr Systems, Inc.
2830 NW 41 St. Bldg M
Gainesville, FL 32606
800-BARRSYS or 904-371-3050

CIRCLE 292 ON READER SERVICE CARD

Utilities

RECORD/PLAYBACK KEYSTROKES

Lost your data due to power failure, accident or frozen keyboard? Recover lost keystrokes with TOTAL RECALL. Also demonstrate/test other programs, see what you typed yesterday, create/run tutorials of other programs, and run interactive programs in batch. PC/XT/AT/Compat. $79 VISA/MC. 30 Day $$$ back guar.
COMPUTER FOUNDATIONS
2604 Elmwood Avenue, Suite 320
Rochester, NY 14618
(716) 586-9756

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

UTILITIES

Tech Marketplace . . .
Listings are grouped by category and sold by column inches.
Call (212) 503-5115

UTILITIES Category
continued on next page

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

UTILITIES

WORD PROCESSING

MISCELLANEOUS
Category
begins on next page

Tech Marketplace...

The comprehensive guide to products and services for the MS DOS market.

Advertising Director
Kathryn J. Cumberlander

Sales Manager
Kathleen F. Lyman

Account Manager
Stanley H. Robinson

Account Representative
Ariane R. Casey

Account Representative
Lee J. Uniacke

Production Manager
Anne R. Brockinton

For additional information call
212-503-5115

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

MISCELLANEOUS

BAR CODING

BUSINESS OPPORTUNITIES

COMPUTER INSURANCE

DATA CONVERSION

To Place Your Ad
Call (201) 503-5115

PRODUCT CATEGORIES

HARDWARE

ACCESSORY CARDS
COOLING DEVICES
COMMUNICATIONS
COMPATIBLES
GENERAL
MASS STORAGE
PERIPHERALS
SECURITY DEVICES
USED EQUIPMENT

SOFTWARE

ARTIFICIAL INTELLIGENCE
BUSINESS
COMMUNICATIONS
DATA BASE MANAGEMENT
EDUCATIONAL
ENGINEERING
EXPERT SYSTEMS
FINANCIAL
GENERAL
GRAPHICS
LANGUAGES
MULTI/USER SYSTEMS
NETWORKING
OPERATING SYSTEMS
PROGRAMMERS TOOLS
PUBLIC DOMAIN
SCIENTIFIC
SECURITY DEVICES
STATISTICS
TERMINAL EMULATION
UTILITIES
WORD PROCESSING

MISCELLANEOUS

ACCESSORIES
BAR CODING
BUSINESS OPPORTUNITIES
COMPUTER INSURANCE
DATA CONVERSION
PUBLICATIONS
SUPPLIES

FOR ADDITIONAL INFORMATION CALL (212) 503-5115

Advertising Director
Kathryn J. Cumberlander

Sales Manager
Kathleen F. Lyman

Account Manager
Stanley H. Robinson

Account Representative
Ariane R. Casey

Account Representative
Lee J. Uniacke

IBM and COMPATIBLE PC's

RS#	PRODUCT	ADVERTISER	PAGE
116	ALR 386/2	Adv. Log. Res.	Inside Back Cover
199	Systems	AST Research	113
*	Deskpro 386/20, Portable 386	Compaq	98 & 99
172	386 XT	Quadram	37

MULTIFUNCTION/MEMORY CARDS

RS#	PRODUCT	ADVERTISER	PAGE
128	16 Megabyte Extend. Memory	Micron Technology Inc.	10

ACCELERATOR BOARDS

RS#	PRODUCT	ADVERTISER	PAGE
145	Monoputer	Microway	38

DESKTOP PUBLISHING

RS#	PRODUCT	ADVERTISER	PAGE
129	NROFF/PC	Elan	165

MASS STORAGE HARDWARE

RS#	PRODUCT	ADVERTISER	PAGE
149	9 Track Tape System	A.K. Systems	155
111	9 Track Tape System	Catamount Corporation	154
133	Nine Track Tape	Flagstaff Engineering	121
108	"LEO"	Innovative Data	152
185	9 Track Hardware	Overland Data, Inc.	118
176	Lan Star Sub-System	Storage Dimensions	19
155	Telebyte Tape Drives	Telebyte	176

PRINTERS-PLOTTERS

RS#	PRODUCT	ADVERTISER	PAGE
243	Tektronix Printers	Tektronix Inc.	28

INPUT SOFTWARE & HARDWARE

RS#	PRODUCT	ADVERTISER	PAGE
113	The Transterm 5 Terminal	Computerwise, Inc.	152

MODEMS

RS#	PRODUCT	ADVERTISER	PAGE
238	V Series 9600/2400	Hayes Microcomputer	96 & 97

DATA ACQUISITION

RS#	PRODUCT	ADVERTISER	PAGE
208	Catalogue	Interactive Microware	154

MONITORS

RS#	PRODUCT	ADVERTISER	PAGE
101	Viking 10	Moniterm Corp.	135
202	Laser View	Sigma Designs Inc.	78

MICRO-MINI MAINFRAME COMMUNICATIONS

RS#	PRODUCT	ADVERTISER	PAGE
122	IDEAcomm 5251/mc	IDEAssociates	107

LOCAL AREA NETWORKS

RS#	PRODUCT	ADVERTISER	PAGE
165	3-N-1	Attachmate Corporation	90
140	LAN Product	Businessland	12 & 13
109	NetCommander	Digital Products	35
106	Lan Report #8	Novell Inc.	162
196	LANLink	Software Link	46
127	TOPS	TOPS, A Sun Microsystems Co.	21
246	Data Gate	Wall Data Inc.	49

OTHER COMMUNICATION HARDWARE

RS#	PRODUCT	ADVERTISER	PAGE
110	Digiboard COM/XI	Digiboard Inc.	153

COMMUNICATION SOFTWARE

RS#	PRODUCT	ADVERTISER	PAGE
167	Crosstalk	Crosstalk Comm.	Back Cover
230	"Specterm"	Meridian Technology	159

OTHER SOFTWARE

RS#	PRODUCT	ADVERTISER	PAGE
176	Lan Star Sub-System	Storage Dimensions	19

APPLICATION SOFTWARE

RS#	PRODUCT	ADVERTISER	PAGE
219	Quic-PROS	QNE International	158

LANGUAGES

RS#	PRODUCT	ADVERTISER	PAGE
251	Pascal, Basic, Prolog	Borland Intl.	Inside Front Cover
160	Lattice Services	Lattice, Inc.	40
229	Modula 2	LOGITECH Inc.	42
263	Personal Rexx	Mansfield Software	174
206	Let's C 4.0	Mark Williams Company	119
120	University	Microsoft Corporation	143
150	MASM	Microsoft Corporation	44
130	Brief	Solution Systems	14
207	True Basic	True Basic Inc.	120

PROGRAMMER'S TOOLS

RS#	PRODUCT	ADVERTISER	PAGE
105	ADA Compiler	Alsys, Inc.	34
203	PC Probe	Atron	26
102	Turbo C Tools	Blaise Computing Inc.	18
254	Turbo C	Borland Intl.	Inside Front Cover
*	Vitamin C	Creative Programming	156
221	Concurrent DOS	Digital Research	6 & 7
119	C-Tree, R-Tree	FairCom	22
139	ProBas	Hammerly Computer Services	32
113	Flowchart	Haventree Software	36
125	Epsilon	Lugaru Software	31
148	C-Tree Query	Machine Independent	131
259	386 Compiler	Metaware	122
191	Screenio	Norcom	148
210	B Trieve	Novell Dev. Div.	8
222	Opt-Tech Sort, Scroll, Recall	Opt-Tech Data Processing	4
104	Grasp	Paul Mace Software	110
233	Programmer's Tools	PC Brand	68–71
114	CBTree	Peacock Software	170
215	Periscope	Periscope Company	5
183	The Norton Editor	Peter Norton Computing	50 & 51
188	Plott 88	Plotworks, Inc.	163
144	PVCS	Polytron Corp.	16
152	"Keep Out" Slate System	Symmetry Group	141
154	Sortex	Systemat Corporation	20
115	Windows for Data	Vermont Creative	17
132	UI Programmer	Wallsoft Systems Inc.	144

SOFTWARE UTILITIES

RS#	PRODUCT	ADVERTISER	PAGE
191	Screenio	Norcom	148
168	Radix Screen Mgmt. System	Radix Systems Inc.	163

GRAPHICS SOFTWARE

RS#	PRODUCT	ADVERTISER	PAGE
107	GraphiC and VTEK	Scientific Endeavors Corp.	155

DATA MANAGEMENT SOFTWARE

RS#	PRODUCT	ADVERTISER	PAGE
*	Magic PC	Aker Corporation	160
224	Clipper	Nantucket Corporation	94
*	Professional Oracle	Oracle Corp.	11
*	DB Vista/DB Query	Raima Corp.	64
237	ZIM	Zanthe	81–83

OPERATING SYSTEMS

RS#	PRODUCT	ADVERTISER	PAGE
253	OS/286 and OS/386	AI Architects	88
174	MKS Tool Kit, AWK	Mortice Kern Systems, Inc.	140
137	DESKview	Quarterdeck Office System	111

HARDWARE DIAGNOSTIC TOOLS

RS#	PRODUCT	ADVERTISER	PAGE
103	H Test/H Format	Paul Mace Software	29

OTHER SERVICES

RS#	PRODUCT	ADVERTISER	PAGE
112	Novell-Developers Conference	Novell Inc.	76
223	Breakthru 286	Personal Computer Support	109
166	Validation Suite	Plum Hall	66

MAIL ORDER

RS#	PRODUCT	ADVERTISER	PAGE
147	Mail Order	Byte Connection, Inc.	180
*	Mail Order	Microway	178
247	Mail Order	Precision Data Products	180
175	Mail Order	Programmer's Connection	23–25
173	Mail Order	Programmer's Paradise	179
162	Mail Order	Programmer's Shop	33
151	Mail Order	Programmer's Shop	171
240	Mail Order	Scantel Systems Ltd.	180

SECURITY DEVICES

RS#	PRODUCT	ADVERTISER	PAGE
240	Software Sentinel	Rainbow Technologies, Inc.	180
189	The Block	Software Security	30

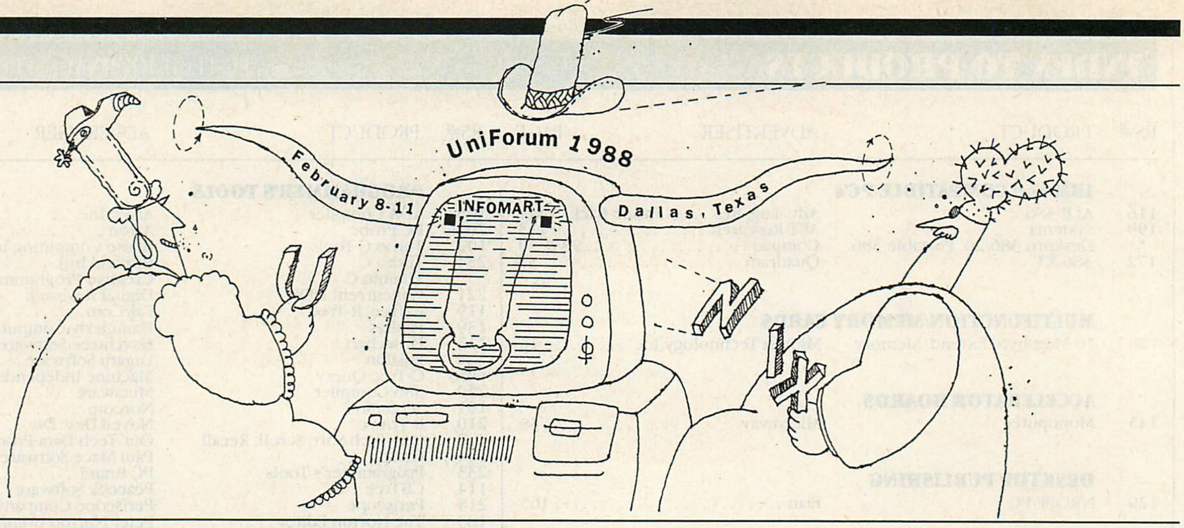


ILLUSTRATION • DAVID POVILAITIS

FEBRUARY

February 2–5
International Conference on Data Engineering
Los Angeles, CA (IEEE-CS) *Contact:* Benjamin W. Wah, Dept. of EE and CE, University of Illinois, Urbana, IL 61801; 217/333-3516

February 8–11
UniForum '88
Dallas, TX (The International Network of UNIX Users) *Contact:* UniForum '88, 2400 E. Devon Avenue, Suite 205, Des Plaines, IL 60018; 800/323-5155; 312/299-3131

February 16–18
DEXPO East '88 Conference
New York, NY (Expoconsul International) *Contact:* Expoconsul, 3 Independence Way, Princeton, NJ 08540; 609/987-9400

February 22–24
Computer Graphics New York
New York, NY (Exhibition Marketing and Management, Inc.) *Contact:* EMM, Inc., 8300 Greensboro Drive, Suite 1110, McLean, VA 22102; 703/893-4545

February 23–25
Computer Science Conference
Atlanta, GA (ACM) *Contact:* Dr. Richard A. DeMillo, Program Chairman, Software Engineering Research Center, Georgia Institute of Technology, Atlanta, GA 30332; 404/894-3180

February 29–March 4
Compcon Spring '88
San Francisco, CA (IEEE-CS) *Contact:* Hasan AlKhatib, Dept. of EECS, U. of Santa Clara, Santa Clara, CA 95053; 408/554-4485

MARCH

March 3–4
LISP: Expert Systems Tools
Atlanta, GA (Georgia Institute of Technology) *Contact:* Deidre Mercer, Education Extension Services, Georgia Tech, Atlanta, GA; 404/894-2547

March 7–10
Computer Workstations
Santa Clara, CA (IEEE-CS) *Contact:* Patrick Mantey, 335A Applied Science Bldg., Dept. of Computer Engineering, U. of California at Santa Cruz, Santa Cruz, CA 95060; 408/429-2158

March 8–10
Technical Conference for MIS/DP Professionals
New York, NY (Cahners Exposition Group) *Contact:* Cahners Exposition Group, 999 Summer Street, Stamford, CT 06905; 203/964-0000

March 8–10
Southcon '88 Electronic Show and Convention
Orlando, FL (IEEE and ERA) *Contact:* Electronic Conventions Management, 8110 Airport Blvd., Los Angeles, CA 90045; 213/772-2965

March 8–11
International Symposium on Digital Communications
Zurich, Switzerland (IEEE-CS) *Contact:* Secretariat IZS 88, c/o P. Gunzburger, Hasler AG, TDS, Belpstrasse 23, CH-3000, Bern 14, Switzerland; 41-31-632808

March 10–11
APPC Communication
San Francisco, CA (Systems Technology Forum) *Contact:* Sherry Armstrong, Seminar Coordinator, Systems Technology Forum, 10201 Lee Highway, Suite 150, Fairfax, VA 22030; 800/336-7409; 703/591-3666

March 14–17
National Conference on Ada Technology
Washington, DC (U.S. Department of Defense) *Contact:* Al Rodriguez, U.S. Army Communications-Electronics Command, Fort Monmouth, NJ 07703; 201/532-4725

March 20–24
NCGA Annual Conference and Exhibition
Anaheim, CA (National Computer Graphics Association) *Contact:* NCGA, 2722 Merrilee Drive, Suite 200, Fairfax, VA; 703/698-9600

March 21–23
Computer Standards Evolution: Impact and Imperatives
Arlington, VA (IEEE-CS) *Contact:* Computer Standards Conference, IEEE, 1730 Massachusetts Avenue, NW, Washington, DC 20036-1903; 202/371-0101

March 21–25
World Users Conference
Los Angeles, CA (MacNeal-Schwendler Corporation) *Contact:* MacNeal-Schwendler Corporation, 815 Colorado Blvd., Los Angeles, CA 90041; 213/258-9111

March 28–31
World Congress on Computing
Chicago, IL (Interface Group) *Contact:* The Interface Group, Inc., 300 First Avenue, Needham, MA 02194; 617/449-6600

March 29–31
Conference on Optical Storage of Documents and Images
Washington, DC (Rothchild Consultants) *Contact:* Rothchild Consultants, 256 Laguna Honda Blvd., San Francisco, CA 94116-1496; 415/681-3700

APRIL

April 11–13
Computer Networking Symposium
Arlington, VA (IEEE-CS) *Contact:* George K. Chang, 6 Corporation Place, Piscataway, NJ 08854; 201/699-3879

April 11–15
International Conference on Software Engineering
Raffles City, Singapore (IEEE-CS, NCB, and ACM) *Contact:* Tan Chin Nam, 71 Science Park, Singapore 0511; 65/772-0200

April 11–15
COMPEURO '88
Brussels, Belgium (IEEE-CS) *Contact:* Jacques Tiberghien, VRIJE Universiteit Brussels, Pleinlaan 2, 1050 Brussels, Belgium; 32-2-641-29-05

April 25–28
International Conference on Expert Database Systems
Tysons Corner, VA (George Mason University) *Contact:* Edgar H. Sibley, George Mason University, ICSE Department, 4400 University Drive, Fairfax, VA 22030; 703/323-2779

MAY

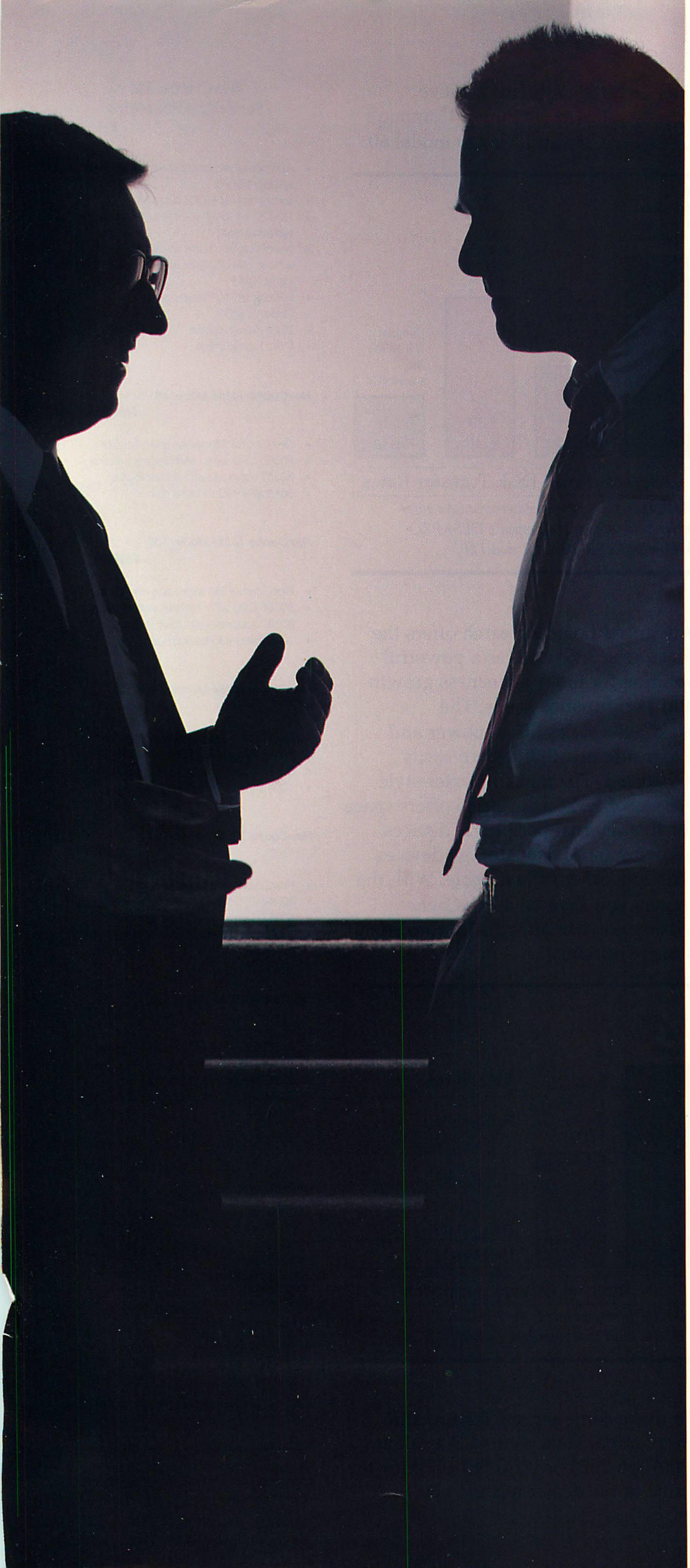
May 5–6
Modeling and Simulation Conference
Pittsburgh, PA (University of Pittsburgh) *Contact:* William G. Vogt, 348 Benedum Engineering Hall, University of Pittsburgh, Pittsburgh, PA 15261; 412/624-9686

May 9–12
COMDEX/Spring '88
Atlanta, GA (The Interface Group) *Contact:* The Interface Group, Inc., 300 First Avenue, Needham, MA 02194; 617/449-6600

May 24–27
Measurement and Modeling of Computer Systems
Santa Fe, NM (ACM SIGMETRICS) *Contact:* Connie Smith, Performance Engineering Services, 1114 Buckman Road, Santa Fe, NM 87501; 505/988-3811

May 30–June 2
International Symposium on Computer Architecture
Honolulu, HI (IEEE-CS, ACM, and SIGARCH) *Contact:* H.J. Siegel, Supercomputing Research Center, 4380 Forbes Blvd., Lanham, MD 20706; 301/731-3700

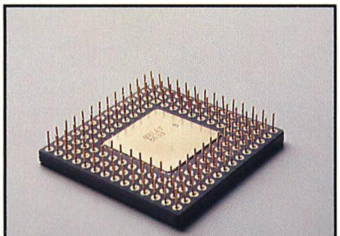
May 31–June 3
1988 National Computer Conference
Los Angeles, CA (AFIPS) *Contact:* American Federation of Information Processing Societies, Preston White Drive, Reston, VA 22091; 703/620-8900



"Spending more money doesn't guarantee that you'll end up with the system you need. Make some serious comparisons of features and prices and then get the most for your money."

POWER

The FlexCache 386 series from Advanced Logic Research



FlexCache 386 series: The performance of 80386 technology with the power of advanced FlexCache architecture.

The philosophy of getting more for your money has become an unbroken tradition with ALR. A tradition that has been recognized by all the major trade journals with excellent reviews.

ALR 386/220, PC magazine's choice as "The Best of 1987"

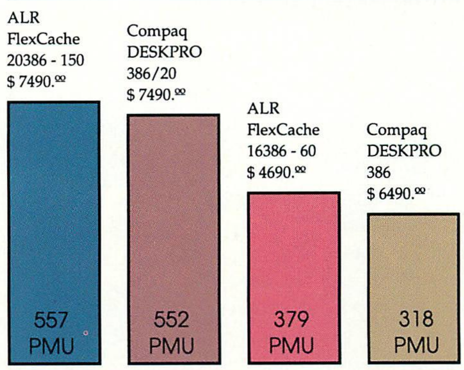
THE BEST OF 1987

Based on the ALR 386/220, PC magazine's choice as "The Best of 1987", ALR extends its product line and introduces the FlexCache 386 series. Now the fastest PCs available, the FlexCache 386 series approach minicomputer proportions and offer two new ways to get the most for your money:

FlexCache 16386 -
a 16MHz, 0-wait-state,
80386/82385 based system.

FlexCache 20386 -
a 20MHz, 0-wait-state,
80386/82385 based system.

Both systems have ALR's advanced FlexCache architecture. The flexible dual bus design provides a wide open, high-speed data channel for up to 60% faster CPU/memory through-put than the IBM PS/2 model 80-071 with the much touted microchannel architecture.



Power Meter Performance Index

FlexCache 386 series edge out Compaq's DESKPRO 386/20 & DESKPRO 386 in CPU/memory aggregate performance test.

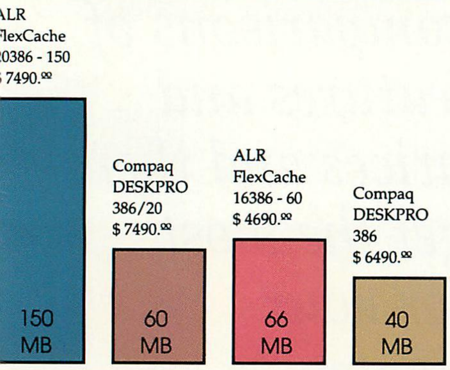
The cache memory controller can eliminate wait-states 95% of the time

by keeping frequently used data close at hand, eliminating the need for the CPU to address main memory. This powerful blend of enhancements allows a FlexCache 16MHz CPU to move data along as fast as many 20MHz CPUs and a FlexCache 20MHz CPU to move data even faster than a Compaq DESKPRO 386/20™.

The FlexCache 386 series comes equiped with the most fixed disk capacity for your money. The FlexCache 16386 has a 66 or 100 megabyte fixed disk.

The FlexCache 20386 will give you an extra 45,000 pages of document disk storage for free.

The FlexCache 20386 comes with either a 100, 150, or 300 megabyte fixed disk. The FlexCache 20386 will give you an extra 45,000 pages of document disk storage for free when you compare it to the performance and price of Compaq's DESKPRO 386/20 model 60.

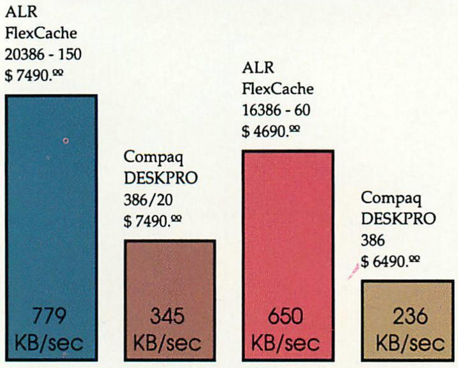


Standard Fixed Disk Capacities

FlexCache 386 fixed disks store thousands more pages of documents than the competition..

FlexCache hard disk controllers transfer a full track of data in one disk revolution (1:1 interleave) instead of several disk revolutions as with (2:1 interleave) most current systems. Full track data transfering plus ESDI (Enhanced Small Device Interface) look-ahead buffering, turns what used to be a data traffic bottleneck into a super high-speed corridor. So

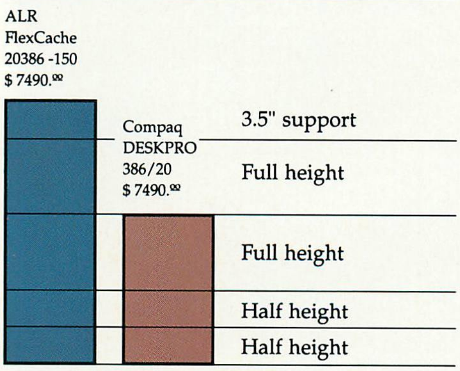
FlexCache 386 systems achieve transfer rates twice that of Compaq's DESKPRO 386 and 386/20 model 60.



Core Test Fixed Disk Transfer Rates

FlexCache 386 systems achieve transfer rates of up to twice that of Compaq's DESKPRO 386 & DESKPRO 386/20 model 60.

Advanced Logic Research offers the FlexCache 386 series as a powerful solution for today's business growth and performance needs. The FlexCache series offers power and expansion possibilities not easily exhausted. The minicomputer-style chassis of FlexCache 20386 offers space for five internal peripheral devices, allowing more data storage devices than any other PC available. With the money you save on a FlexCache system you can afford additional data storage options.



Internal Device Support

With the future in mind, the FlexCache 20386 is built to accomodate growth.

To make some serious feature and cost comparisons give ALR a call at (800) 366-2574 or (714) 581-6770 for the name of the dealer nearest you.

FlexCache 386 Series Specifications

- ALR designed and proven multi-layer system board
- Socketed for 80387 support
- 1MB 32 bit RAM, expandable to 2MB on system board
- 0-wait-state cache memory controller with its own 32KB of high-speed (35 ns) static RAM
- Enhanced 101 keyboard
- Phoenix BIOS
- Dual drive support
- OS/2 compatible

FlexCache 16386 Model 60 **...$4690.00**

- FlexCache 386 series specifications
- 80386 CPU with 16MHz system clock
- 80387 support with 16MHz clock
- 66Mbyte <30ms hard disk

FlexCache 16386 Model 100 **...$5690.00**

- FlexCache 386 series specifications
- 80386 CPU with 16MHz system clock
- 80387 support with 16MHz clock
- 100Mbyte <30ms hard disk

FlexCache 20386 Model 100 **...$6490.00**

- FlexCache 386 series specifications
- 80386 CPU with 20MHz system clock
- 80387 support with 20MHz clock
- 100Mbyte <30ms hard disk

FlexCache 20386 Model 150 **...$7490.00**

- FlexCache 386 series specifications
- 80386 CPU with 20MHz system clock
- 80387 support with 20MHz clock
- 150Mbyte <23ms, track buffered, ESDI hard disk

FlexCache 20386 Model 300 **...$9990.00**

- FlexCache 386 series specifications
- 80386 CPU with 20MHz system clock
- 80387 support with 20MHz clock
- 300Mbyte <20ms, track buffered, ESDI hard disk

Advanced Logic Research, Inc.
10 Chrysler, Irvine, CA 92718
714-581-6770
FAX: 714-581-9240
Telex: 5106014525,
Answer back Advanced Logic

From Asia or Europe call:
ALR/Wearnes Technology
(65) 2592521 (Singapore)
Telex: RS38113 WRNTEC

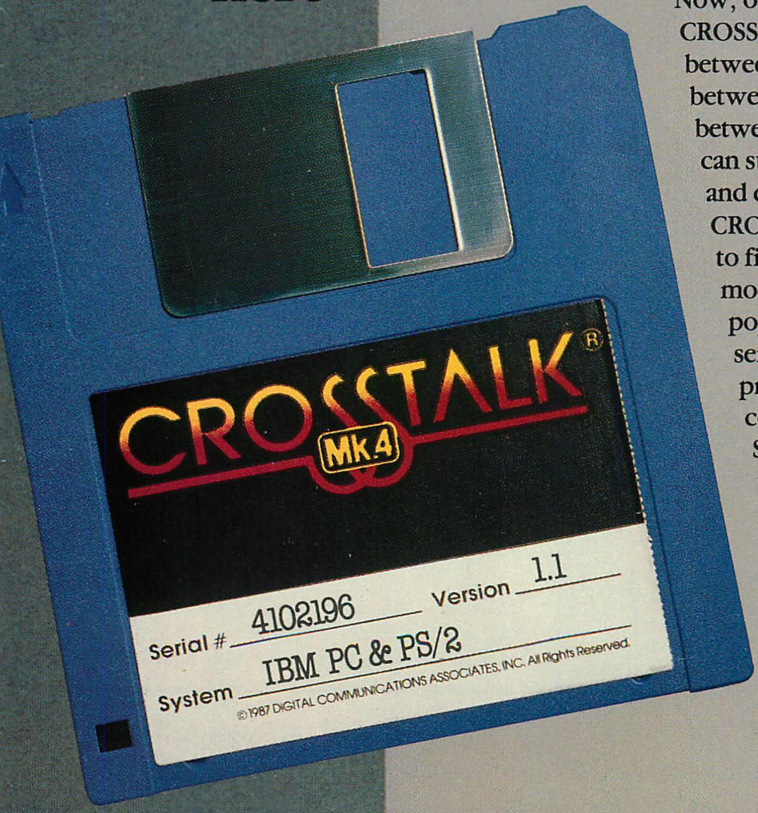
Circle No. 116 for Re-seller, No. 121 for End-user.

IBM Spoken Here

and here

and here

and here



Whatever dialect of IBM you need to speak, CROSSTALK® Mk. 4 makes the connection.

Now, one program does the job that used to require several. CROSSTALK® Mk. 4 allows high-speed direct communications between PCs and minicomputers, or (with an IRMA™ board) between your PC and an IBM Mainframe, or (with Smart Alec™) between your PC and IBM System 3x's. If you like, CROSSTALK can support all of these sessions (and others) simultaneously, and display each session in its own window.

CROSSTALK Mk. 4 emulates all the terminals you're likely to find useful. That includes IBM 3101 (page and character modes), IBM 525x, IBM 529x, IBM 327x, as well as many popular async terminals like the DEC VT100 and VT220 series. CROSSTALK Mk. 4 includes the powerful CASL™ programming language, which allows you to automate communications applications quickly and easily.

So if you're used to thinking of CROSSTALK just to use with a modem, you're missing some important connections. Ask your dealer for details, or write:

dca® Digital Communications Associates, Inc.
1000 Holcomb Woods Parkway / Roswell, Georgia 30076
1-800-241-6393

CROSSTALK® COMMUNICATIONS

CIRCLE NO. 167 ON READER SERVICE CARD